SCHOOL OF CIVIL ENGINEERING

JOINT HIGHWAY RESEARCH PROJECT

JHRP-76-16

FINLIN USER'S MANUAL

M. B. Roy

PURDUE UNIVERSITY
INDIANA STATE HIGHWAY COMMISSION

FINLIN USER'S MANUAL

TO: J. F. McLaughlin, Director
Joint Highway Research Project

May 5, 1976

Project: C-36-62F

FROM: H. L. Michael, Associate Director
Joint Highway Research Project

File: 9-8-6

The attached "FINLIN User's Manual" is provided for the computer program FINLIN developed in the HPR Part II Research Study "Performance of Pipe Culverts Buried in Soil". Its development and formulation is reported in the Interim Report "Predicting Performance of Pipe Culverts Buried in Soil", M. B. Roy, JHRP-76-15, May 1976.

This Report and its companion volume, JHRP-76-15, were presented to the JHRP Board at its meeting on May 5, 1976, and accepted as fulfillment of the objectives of Phase I of the Study. It is now forwarded for review, comment and similar acceptance by ISHC and FHWA.

Respectfully submitted,

Harold L. Michael

Harold L. Michael
Associate Director

HLM:ms

cc:
W. L. Dolch
R. L. Eskew
G. D. Gibson
W. H. Goetz
M. J. Gutzwiller
G. K. Hallock
D. E. Hancher
M. L. Hayes
K. R. Hoover
G. A. Leonards
C. W. Lovell
R. D. Miles
P. L. Owens
G. T. Satterly
C. F. Scholer
M. B. Scott
K. C. Sinha
L. E. Wood
E. J. Yoder
S. R. Yoder

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
JHRP-76-16		

4. Title and Subtitle

FINLIN USER'S MANUAL

5. Report Date

May 1976

6. Performing Organization Code

7. Author(s)

M. B. Roy

8. Performing Organization Report No.

JHRP-76-16

9. Performing Organization Name and Address

Joint Highway Research Project
Civil Engineering Building
Purdue University
W. Lafayette, Indiana 47907

10. Work Unit No.

11. Contract or Grant No.

HPR-1(12) Part II

12. Sponsoring Agency Name and Address

Indiana State Highway Commission
State Office Building
100 North Senate Avenue
Indianapolis, Indiana 46204

13. Type of Report and Period Covered

User's Manual
Computer Program
FINLIN

14. Sponsoring Agency Code

15. Supplementary Notes

Prepared in cooperation with the U.S. Department of Transportation, Federal Highway Administration.
From the study, "Performance of Pipe Culverts Buried in Soil".

16. Abstract

This is a User's Manual for the Computer Program FINLIN (Finite element, Isoparametric, Non-Linear with Interaction and No-tension). It is a finite element computer program for analysis of flexible pipe culverts buried in soil. Details of mathematical formulations, developments of different element properties, and related information can be found in the Interim Report "Predicting Performance of Pipe Culverts Buried in Soil", M. B. Roy, JHRP-76-15, May 1976.

17. Key Words

Pipe Culverts; Design of Pipe Culverts; Finite Element Analysis; Pipe Culverts in Soil

18. Distribution Statement

No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	77	

Form DOT F 1700.7 (8-69)

User's Manual

FINLIN USER'S MANUAL

by

M. B. Roy
Graduate Instructor in Research

Joint Highway Research Project

Project No.: C-36-62F

File No.: 9-8-6

Prepared as Part of an Investigation

Conducted by

Joint Highway Research Project
Engineering Experiment Station
Purdue University

in cooperation with the

Indiana State Highway Commission

and the

U. S. Department of Transportation
Federal Highway Administration

This User's Manual is for the computer program for analysis of flexible pipe culverts buried in soils and reported in Interim Report "Predicting Performance of Pipe Culverts Buried in Soil", M. B. Roy, JHRP No. 15, May 1976.

Purdue University
West Lafayette, Indiana
May 5, 1976

User's Manual

RIPLEM USER'S MANUAL

to

by

[illegible]
Graduate Instructor in Research

Joint Highway Research Project

Project No.: [illegible]

File No.: [illegible]

Prepared as Part of an Investigation

Conducted by

Joint Highway Research Project
Engineering Experiment Station
Purdue University

in cooperation with the

Indiana State Highway Commission

and the

U. S. Department of Transportation
Federal Highway Administration

The contents of this report reflect the views of the author who is responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

This User's Manual is for the computer program for analysis of flexible pipe culverts buried in soils and reported in Interim Report "Predicting Performance of Pipe Culverts Buried in Soil", [illegible], JHRP No. 15, May 1976.

Purdue University
West Lafayette, Indiana
May 1, 1976

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I. INTRODUCTION

FINLIN (Finite element, Isoparametric, Non-Linear with Interaction and No-tension) is a finite element computer program for analysis of flexible pipe culverts buried in soil. Curved bar segments have been used to simulate flexible pipes. A zero thickness special type of frictional element is used to simulate the occurrence of slip between pipe and soil. Isoparametric, linear-strain triangular elements are used to represent the soil. Non-linear, anisotropic, state of stress-dependent soil properties expressed in terms of octahedral stresses are used. Several option commands permit realistic analysis of a culvert problem in two dimensions. Incremental stages of construction, controlled slip between soil and pipe and no-tension analysis can be performed. The program has been written in FORTRAN IV source language using hard and soft wire facilities available in the CDC computer system. In this report, the procedure for preparation of input data for a problem will be discussed; a complete listing of the source program is included. Details of mathematical formulations, developments of different element properties, and related information can be found in the Interim Report - "Predicting Performance of Pipe Culverts Buried in Soil", by M. B. Roy, Purdue University, May, 1976, JHRP-76-15.

In any finite element analysis the first step is to bound the problem by a set of finite boundaries with appropriate boundary conditions. Then each region of different materials has to be distinguished. In the next step, each zone is subdivided into a number of characteristic finite elements maintaining continuity at the boundary between two elements. Then nodes and elements are numbered in sequence. Required material properties for every element need to be defined. In the case of nonlinear analysis, where material properties change depending upon the state of stress, parameters which govern the variation of properties need to be defined. Also, construction in layers, load application in increments, limited shear or slip in the pipe-soil inter-

CHAPTER I. INTRODUCTION

FINLIN (finite element, [illegible]) with interaction and [illegible]) is a finite element computer program for analysis of flexible pipe culverts buried in soil. Several new features have been added to [illegible] a zero thickness special type of "interface" elements [illegible] to simulate the occurrence of slip or [illegible]. Isoparametric [illegible] quadrilateral or triangular elements are used to represent the soil. Non-linear, [illegible] dependent soil properties [illegible] elements are used. [illegible] analysis of a [illegible] [illegible] of consolidation [illegible] and no-tension analysis [illegible] been written [illegible] [illegible] In this report, [illegible] the [illegible] culvert program [illegible] formulations, [illegible] and related information [illegible].

[illegible] Predicting Performance of [illegible] [illegible] Research Record, [illegible]

Usually finite element analysis [illegible] based on the problem by a set of [illegible] the boundary conditions. Then each region [illegible] materials has to be distinguished. [illegible] zone is represented using a number of [illegible] finite elements maintaining continuity at the nodes between the elements. Then nodes and elements are numbered in sequence. Required material properties for each element need to be defined. In the case of nonlinear analysis where material properties change depending upon the state of stress, parameters which govern the variation of properties need to be defined. Also, construction in layers, load application in increments, limited shear or slip in the pipe-soil inter-

action, and no-tension in soil must be accommodated.

Preparation of data for a real problem should take advantage of the fact that:

1. The boundaries need only define the areas of primary interest, and obvious features, like symmetry, should be recognized.
2. Smaller sized elements are needed in zones of maximum interest and/or high stress gradients; larger elements may be used elsewhere.
3. Each region is divided into an appropriate number and type of elements whose node points are numbered sequentially . The cost of a solution depends heavily on the numbering sequence of nodes even if the total number of nodes and elements remain the same. A useful general rule is to minimize the maximum difference between the highest and lowest node numbers in an element.

The next step is selection of the type of analysis, such as number of construction layers, number of increments of load in a given layer, interaction properties, no-tension in soil and other similar decisions. All information has to be digitized and checked for correctness.

actions, and Washington in-call must be accommodated.

Preparation of data for a [illegible] should take advantage of the fact that

1. The parameters used only define the group of airfoils [illegible] and various [illegible] geometry, should be [illegible].
2. [illegible] a [illegible] in [illegible] larger elements [illegible].
3. [illegible] and [illegible] in [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] and [illegible]

The [illegible]

[illegible]

CHAPTER II. COMPUTER PROGRAM

1. Types of Finite Elements

Several types of finite elements have been used in the program FINLIN. Description of each type follows.

Type I, Curved Bar Element

Segments of a ring have been used to represent flexible pipe, which has small thickness compared to the radius. Figure 1 shows a typical Type I element with two nodes, each node having three degrees of freedom - radial, tangential and rotational. The radius of the pipe, its stiffness EI (young's modulus times moment of inertia), and the nodal coordinates need to be defined. The position of the center of curvature is also necessary.

Type II, Interaction Element

This is a zero thickness, rectangular element with four nodes, each node having two degrees of freedom in the normal and tangential directions. Figure 2 shows a typical interaction element where the coordinates of nodes 1 and 4 (and 2 and 3) are initially the same. Coordinates of all four nodes, stiffness values in normal and tangential directions, E_n and E_s respectively, need to be defined. The program will modify the values of E_n and E_s according to the state of stress in these elements.

For both Type I and Type II elements, the program will perform the necessary coordinate transformation depending upon their position and orientation.

Type III, Isoparametric Triangular Element

This type of triangular element has three corner nodes and three intermediate (midpoint) nodes, each node having two degrees of freedom (Figure 3). The face 1-4-2 can be a

CHAPTER II. COMPUTER PROGRAM

I. Types of Finite Elements

Several types of finite elements have been used in the program FINIT[illegible]. Descriptions of each type follow.

Type I. [illegible]

[illegible]

Type II. [illegible]

[illegible]

For each Type I and Type II elements, [illegible] compute the material [illegible] their position and orientation.

Type III. Isoparametric Triangular Element

This type of triangular element has three corner nodes and three intermediate (midside) nodes; each node has two degrees of freedom (Figure 3). The [illegible]

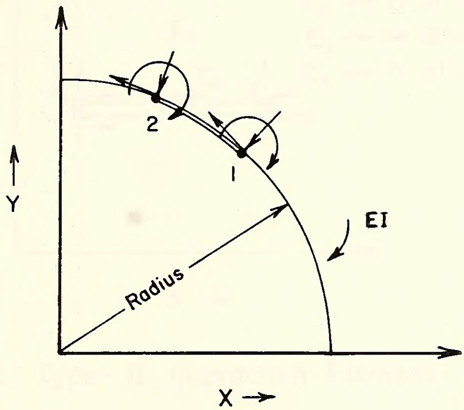


Figure 1. Type - I, Curved Bar Element

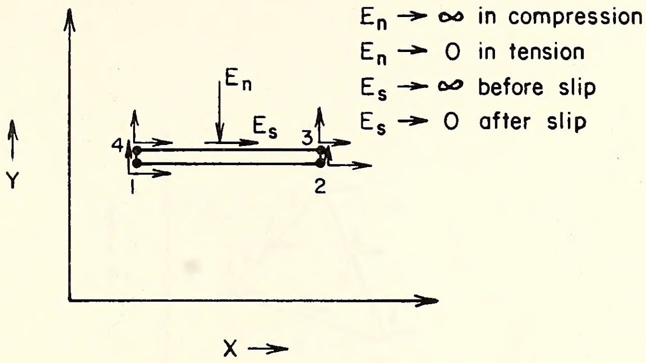


Figure 2. Type - II, Interaction Element

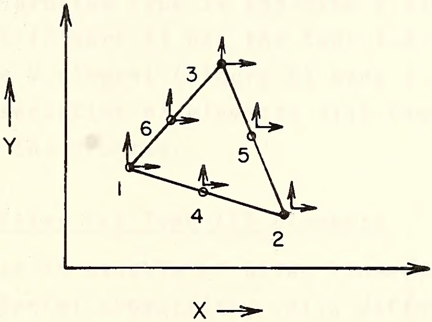


Figure 3. Type-III, Isoparametric Triangular Element

curved boundary. Nodes of the Type III element must be defined in the order shown in Figure 3. Coordinates of only corner nodes need to be defined. This element is used to simulate the soil medium and for thick pipes where rotation at pipe nodes are not significant. Required material properties for this type of element are modulus and poisson's ratio. Depending upon the nature of the material, linear or nonlinear properties can be used.

In case of analysis with no interaction element between pipe and soil, placing Type III element adjacent to Type I elements causes a difficulty in numerical procedure because nodes of 2 and 3 degrees of freedom lie at one point. To eliminate this problem Type IV and Type V elements are used. Type IV element (Figure 4) has the face 1-4-2 adjacent to the pipe. For Type V element (Figure 5) node 3 touches the pipe. Improper representation of elements will cause abnormal termination of the program.

Material Properties for Type III Elements

The program is capable of using linear, nonlinear and anisotropic material properties. Also different types of soils with distinct properties can be used. For linear materials only values of Young's modulus E and Poisson's ratio ν need to be defined. For anisotropic material, the ratio of moduli in vertical to horizontal direction has to be defined. For nonlinear properties, experimental data are directly used. It is to be noted that, in this program tangent modulus and tangent Poisson's ratio values are used for incremental analysis. Also octahedral normal and shear stresses have been used in the formulation. Nonlinear soil properties for any value of stress level and stress ratio are interpolated using cubic spline functions.

It is necessary to convert conventional test data (e.g. Figure 6) to a form which is acceptable to this program, as follows:

corner boundary. Nodes of the mesh are numbered and de- fined in the order shown in Figure 3.3. Coordinates of each corner node need to be defined. Only elements [illegible] simulate the soil medium and the rigid plate. [illegible] of type [illegible] are not illustrated. [illegible] suggests that this type of element [illegible] static. Depending upon the [illegible] nonlinear properties [illegible] can be used.

In case of [illegible] [illegible] and [illegible] [illegible] elements [illegible] [illegible] of 2 and 3 [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible]

[illegible]

[illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible]

It is necessary to convert [illegible] Figure 3.3 [illegible] acceptable [illegible] follows:

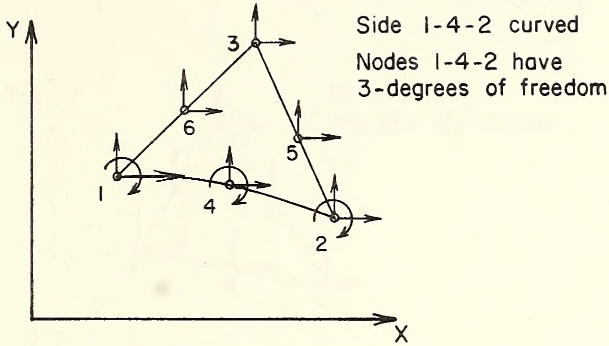


Figure 4. Type - IV, Triangular Element

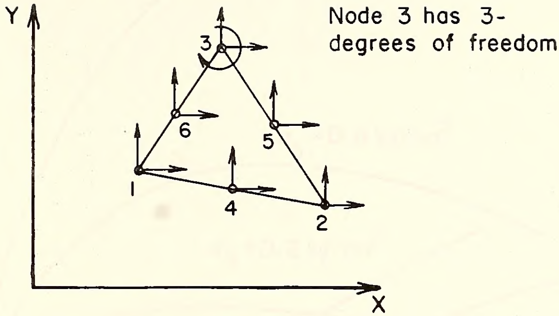


Figure 5. Type - V, Triangular Element

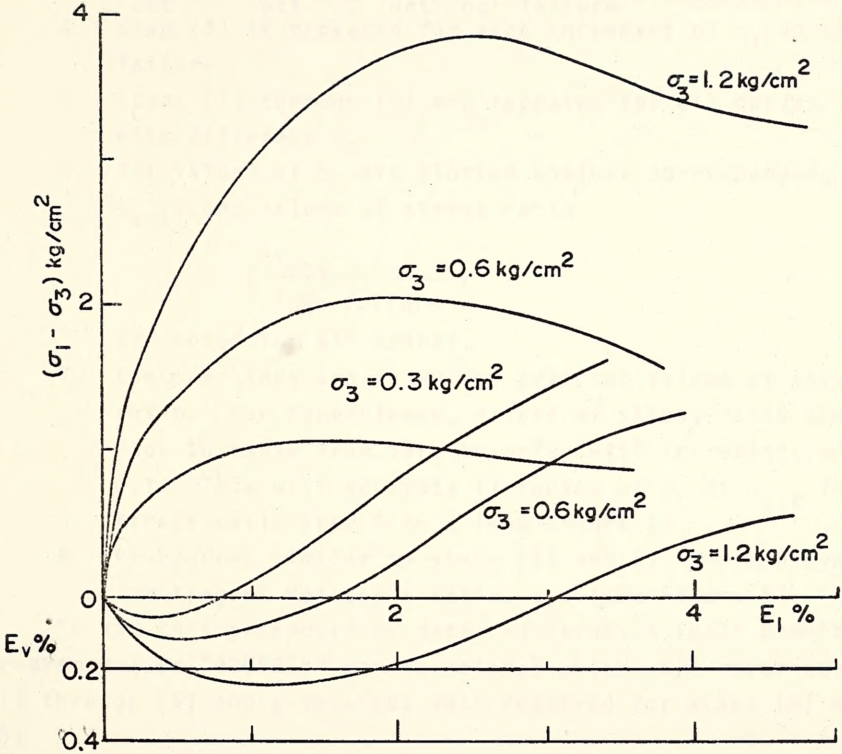


Figure 6, Plane Strain Test on Loose Monterey No.0 Sand (after Lade, 1972)

1. Select successive values of $(\sigma_1-\sigma_3)$, ε_v and ε_1 from the test data, at a given σ_3 starting from zero.
2. Cubic splines are fitted to the data.
3. Small increments of ε_1 are chosen, and values of σ_2, E_t and ν_t are computed from the generalized Hooke's Law. With σ_1, σ_2, and σ_3 known, values of σ_{oct} and τ_{oct} and $\tau_{oct}/\tau_{oct\ failure}$ are calculated.
4. Step (3) is repeated for each increment of ε_1 up to failure.
5. Steps (1) through (4) are repeated for all curves with different σ_3.
6. All values of E_t are plotted against corresponding σ_{oct}, and values of stress ratio

$$\left(\frac{\tau_{oct}}{\tau_{oct\ failure}} \right)$$

   are noted for all points.
7. Contour lines are drawn for selected values of stress ratio. For convenience, values of stress-ratio are made to range from zero to unity with increments of 0.1. This will generate 11 curves of E_t vs σ_{oct} for stress ratio from 0 to 1 (see Figure 7).
8. Procedures similar to steps (6) and (7) are employed for tangent Poisson's ratio, ν_t, (see Figure 8).

To aid this procedure of data reduction, a small computer program called "PROPRTY" can be helpful which will cover steps (1) through (5) and prints out data required for steps (6) and (8).

Description of the Computer Program

The program FINLIN has been divided into several primary sections which are called OVERLAYS. Each OVERLAY performs specific computations and stores the results for future use. Figure 9 shows the principal organization of the program and several OVERLAYS. Each OVERLAY consists of one main program

1. Select successive values of the angle α, and, from the test data, at a strain ϵ_2 starting from zero.
2. Cubic splines are fitted to the data.
3. Small increments of ϵ_2 are chosen, and values of σ_2, ϵ_1, and ϵ_3 are computed from the cubic splines [illegible]. With σ_2, ϵ_1, and ϵ_3 known, values of [illegible] and [illegible] are calculated as follows.
4. Step (3) is repeated for each increment of [illegible] to failure.
5. Steps (1) through (4) are repeated for all values of [illegible] differences [illegible].
6. All values of [illegible] are plotted against [illegible] the [illegible] of [illegible]

   [illegible]

   [illegible]

   are [illegible]
7. [illegible]
   [illegible]
   [illegible]
   [illegible]
8. Procedures given [illegible] for [illegible]

In both this [illegible] program called "[illegible]" [illegible] (1) through (8) and prints out data [illegible]

Organization of the Computer Program

The program CYCLIN has been divided into several subroutines which are called "OVERLAYS". Each OVERLAY performs specific computations and stores the results for future use. Figure 8 shows the principal organization of the program and several OVERLAYS. Each OVERLAY consists of one main program

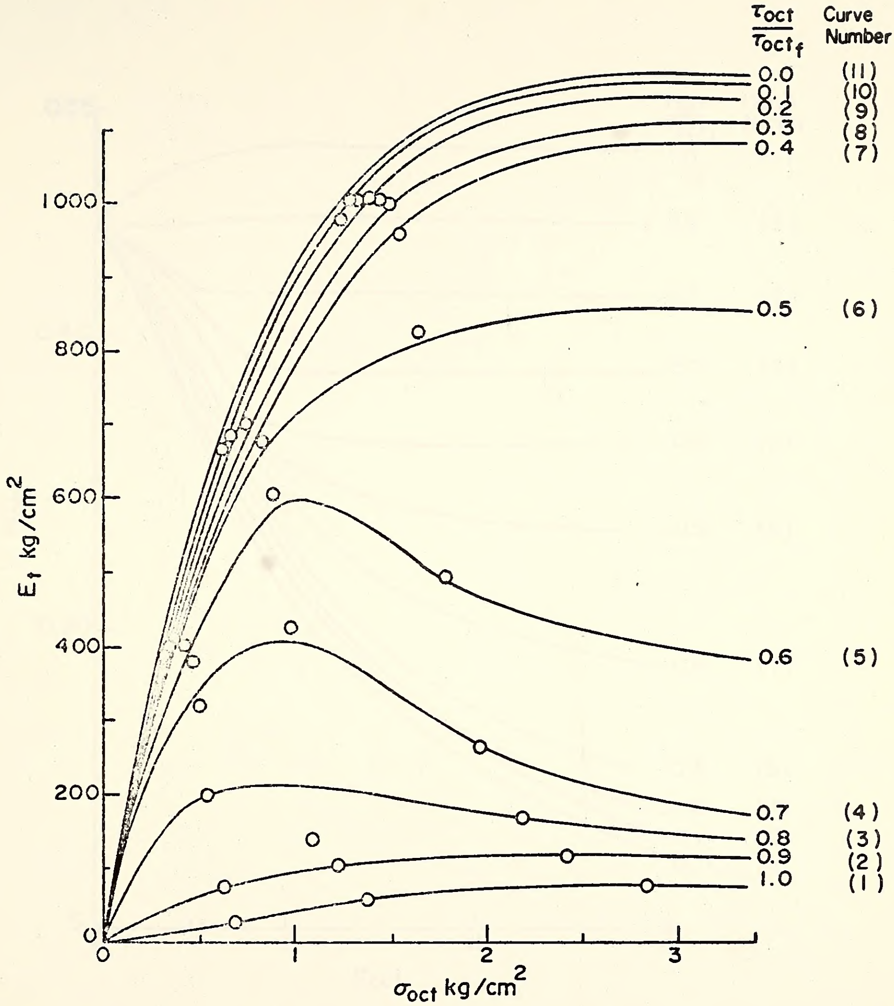


Figure 7, Tangent Modulus vs. Octahedral Normal Stress and Failure Ratio for Loose Sand.

Figure 7. Tangent Modulus vs Octahedral Normal Stress and Failure Ratio for Lucas Sand

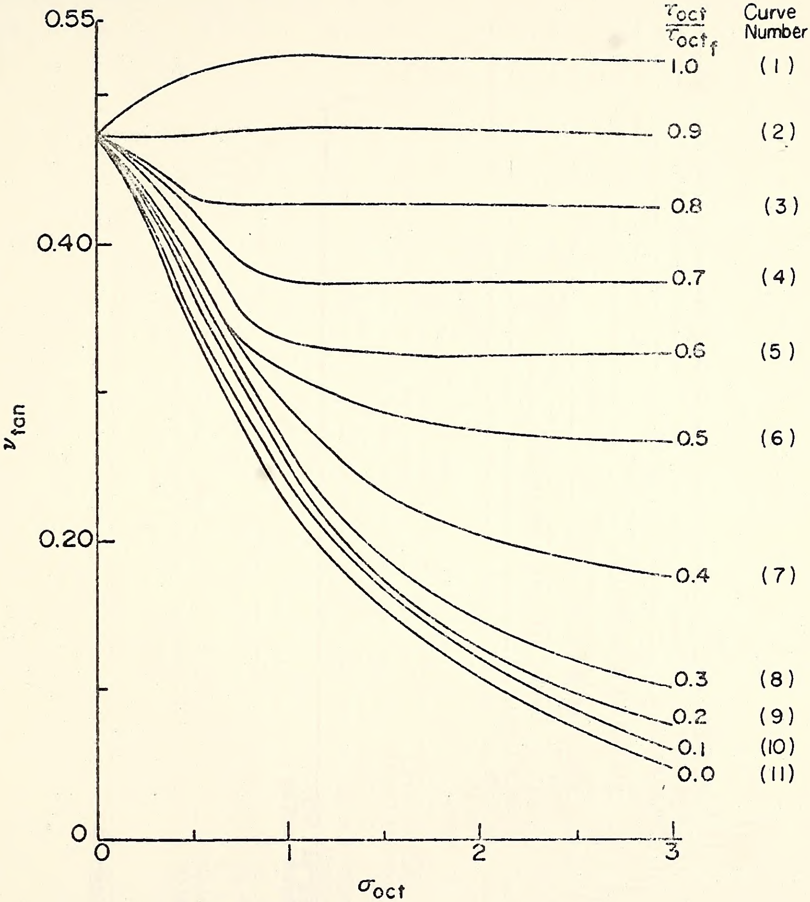


Figure 8, Tangent Poisson's Ratio vs. Octahedral Normal Stress and Failure Ratio for Loose Sand.

Figure 8. Tangent Poisson's Ratio vs Octahedral Normal Stress and Failure Ratio for Loose Sand.

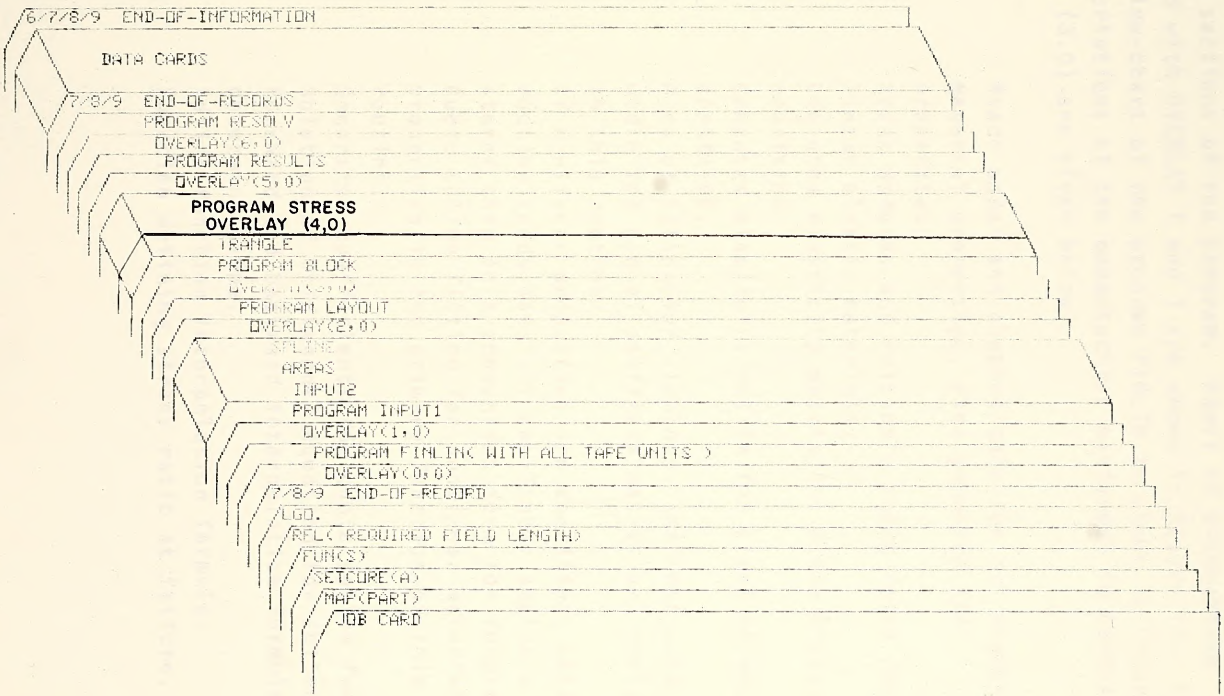


Fig. 9 ARRANGEMENT OF ROUTINES IN "FINLIN"

and several subroutines. OVERLAY 1 and 3 are the two most important sections of the program. Names of subroutines associated with OVERLAY 1 and 3 are shown in Figure 10. A general flow-chart of the program FINLIN is shown in Figure 11. Brief descriptions of the computations performed in OVERLAY (1.0) and (3.0) are given below.

INPUT1 -	Reads nodal and element data for all elements, material properties, pipe geometry and properties.
INPUT2 -	Reads modulus and Poisson's ratio value for linear elastic materials.
PLSTRS -	Computes elasticity matrix for plane stress condition.
PARAMTR -	Computes elasticity matrix for plane strain condition.
AREAS -	Area of triangular elements, and semi-band width for global stiffness matrix are evaluated in this routine.
SPLINE -	If nonlinear properties are specified, this routine reads data for non-linear analysis and stores them in a convenient form for future use.
SPLFIT -	Cubic spline fitting for nonlinear material properties is the primary function of this routine.
COFRIT -	Generates coefficients for cubic spline function.
TRIDGNL -	Solution of tridiagonal equation for spline fit.
FD and BD -	Foreward and backward interpolation formulae to determine slope.
ORDINET -	Spline-function interpolation formula.
FAILURE -	Function defining stress ratio at failure.

and several subroutines. OVERLAY 1 and 2 are the two most important sections of the program. Names of subroutines associated with OVERLAY 1 and 2 are shown in Figure [illegible]. A general flow chart of the program TRACER is shown in Figure [illegible]. Brief descriptions of the subroutines contained in OVERLAY (1,0) and (2,0) are given below:

INPUT1 - Reads model and element data for the electrical material properties plus geometry and boundaries.

INPUT2 - Reads boundary and external electric stress (large electric [illegible]).

MATRIX - Computes [illegible] matrix [illegible] (of [illegible]).

FACTORIZ - Computes [illegible] equations.

ANGLE - [illegible]

SPLINE - [illegible]

SELECT - [illegible]

OUTPUT - [illegible]

INTERP - Calculates quantities [illegible] subroutines forward and backward transformation [illegible] [illegible]

INDIRECT - Spline function interpolation formula

FATIGUE - Function defining stress ratio at failure

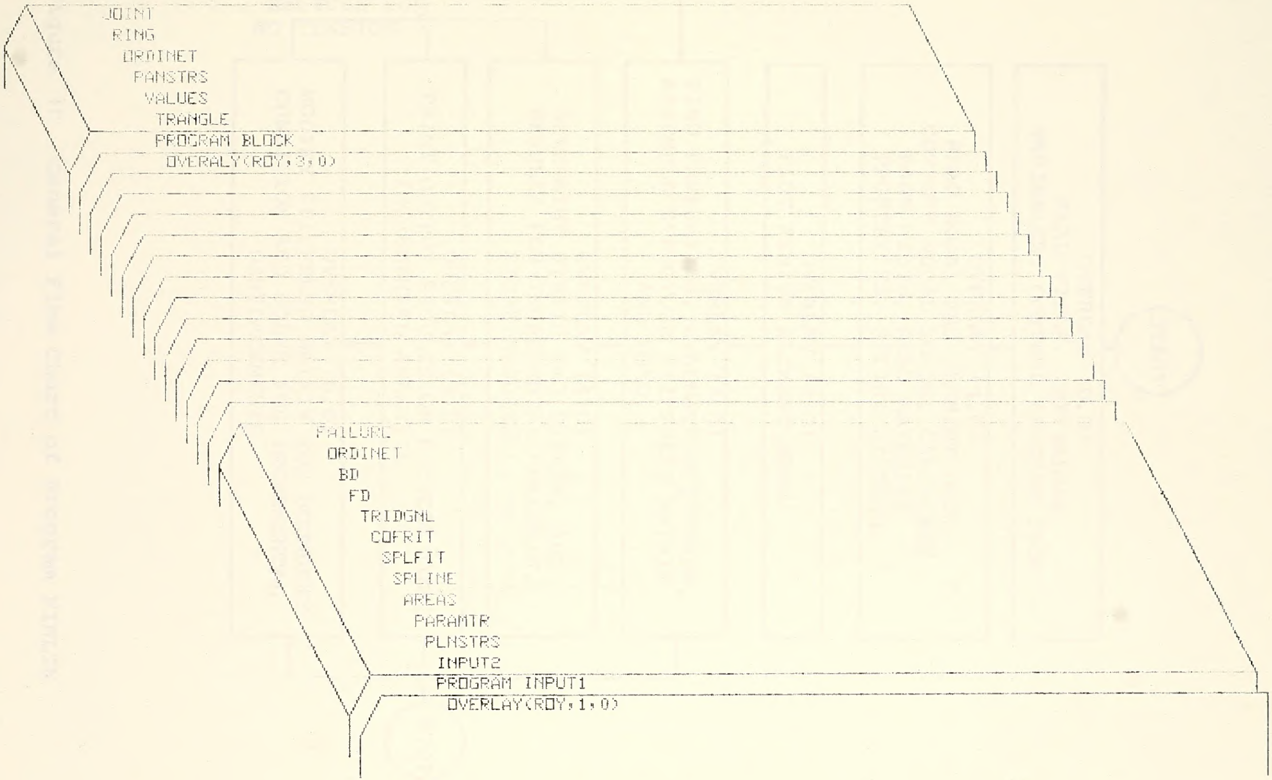


Fig. 10 SUBROUTINES IN OVERLAY (1.0) AND (3.0)

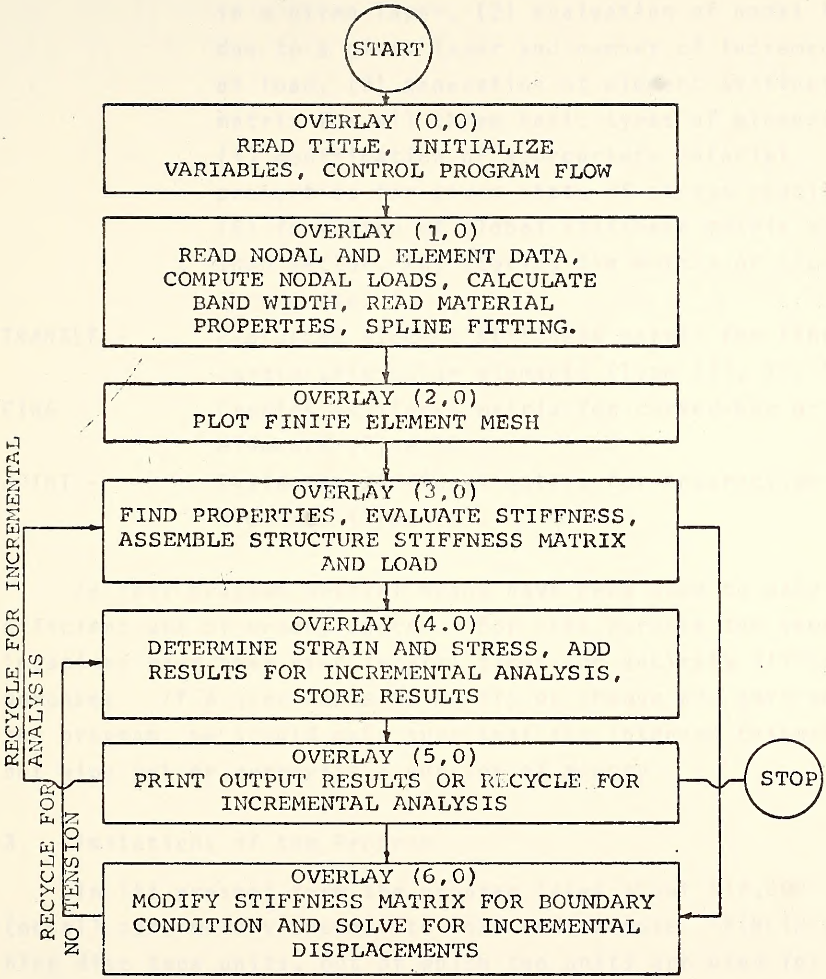


Figure 11 General Flow Chart of Program FINLIN

BLOCK - This is the most important routine in FINLIN. Primary functions are (1) selection of elements in a given layer, (2) evaluation of nodal loads due to a given layer and number of increments of load, (3) generation of element stiffness matrix for all three basic types of elements, (4) modification of appropriate material properties for given state of stress condition, (5) formation of global stiffness matrix and load vector, (6) storing the matrix on tape for future use.

TRANGLE - Evaluates element stiffness matrix for linear-strain-triangular elements (Type III, IV, V).

RING - Creates stiffness matrix for curved-bar or ring elements (Type I).

JOINT - Evaluates stiffness matrix for interaction elements (Type II).

In this program several means have been used to make efficient use of memory spaces. For this purpose the same locations have been used several times for entirely different purposes. If a user wants to modify or change any portion of the program, he should make sure that the intended changes do not wipe out or overwrite a portion of memory.

3. Limitations of the Program

In its present form the program takes about 112,000 (octal) core memory spaces to load and execute. FINLIN uses nine disc tape units, out of which two units are used for INPUT, OUTPUT, five for random access mass storage and two for intermediate storage purposes. Limitations of the program are listed below (they can be modified easily by changing a few cards):

(1) Total number of nodes: (NNODES) = 550
(2) Total number of elements: (NELEMNT) = 250
(3) Different types of materials: (MATRIAL) = 25
(4) Increments of stress ratio in Figure 7 and 8 = 0.1
(5) Maximum number of points for nonlinear material properties = 7
(6) Maximum semi-bandwidth for global stiffness matrix (including diagonal) = 103

However, semi-bandwidth capacity can be modified easily by making changes as follows:

(a) Estimate the maximum semi-bandwidth (including the diagonal term). If it is even, add one and make it odd numbered, which is the final semi-bandwidth (N) of the system. Numerals in the indicated places should at least be N.
(b) OVERLAY (0.0), Line: FLN-36, NSIZE = N
(c) OVERLAY (3.0), Line: BLK-15
DIMENSION A(N,N), ARRAY (N)
(d) OVERLAY (6.0), Line: SOL-10
DIMENSION A(N,2N), B(2N), ARRAY(N)
(e) SUBROUTINE MODIFY, Line: 3
DIMENSION A(N,2N), B(2N).

CHAPTER III. INPUT DATA CARDS FOR PROGRAM "FINLIN"

<u>1st Card Type</u>: TITLE, MESH
FORMAT (13A 6, I2)
One card identifying the problem.

Col. 2-78: alphanumeric description of the problem to be printed in the output.

Co. 79-80: if greater than 0, a plot of finite element mesh is generated by CALCOMP plotter. If zero or blank, no plot is generated.

<u>2nd Card Type</u>: NNODES, NELEMNT, MATRIAL, NPRSR, LAYERS, ISTOP
FORMAT (2I5, 2I3, 2X, I2, 2X, I2)
One card defining problem.

Col. 1-5: NNODES - Total number of node points.

Col. 6-10: NELEMNT - Total number of elements (all types included).

Col. 11-13: MATRIAL - Total number of different soil types.

Col. 14-16: NPRSR - 0

Col. 19-20: LAYERS - Number of construction layers.

Col. 23-24: ISTOP - If zero, linear elastic soil properties, if greater than zero, nonlinear soil properties.

<u>3rd Card Type</u>: NTYPE, GAMA, TYPE
FORMAT (I5, F10.0, 10A6)
Soil type cards, one card for each soil type, total number = MATRIAL

Col. 1-5: NTYPE - Soil-type identification number.

Col. 6-15: GAMA - Unit weight of soil.

Col. 16-75: TYPE - Alphanumeric description of this soil type.

4th Card Type: ANLSIS, DELTA

FORMAT (A6, F5.0)

One card specifying type of analysis and angle of friction between soil and pipe.

Col. 1-6: PLSTRS - For plane-stress analysis.

PLSTRN - For plane-strain analysis.

Col. 7-11: DELTA - Angle of friction between pipe material and soil adjacent to pipe, in degrees.

5th Card Type: Two cards per soil material type, total number = 2X MATRIAL, specifying initial modulus and Poisson's ratio for each soil type.

(a) 1st Card - E

FORMAT (E10.0)

Col. 1-8: E - Initial modulus.

(b) 2nd Card - NUE

FORMAT (F5.0)

Col. 1-5: NUE - Initial Poisson's ratio.

6th Card Type: These cards are required only for nonlinear materials i.e. if ISTOP is greater than zero (in 2nd Card Type). This set of cards is repeated for each nonlinear soil type. If ISTOP = 0 or blank, these cards are not required.

(a) NP, PSI, PHI, ANISO, DELTA, FACTOR

FORMAT (I5, 5F10.0)

One card identifying soil properties.

Col. 1-5: NP - Number of points on each tangent modulus and tangent Poisson's ratio vs. σ_{oct} curve.

Col. 6-15: PSI - Factor defined as

$$\Psi = \frac{\sigma_2}{(\sigma_1+\sigma_3)}$$

Col. 16-25: PHI - Friction angle for soil, in degrees.

Col. 26-35: ANISO - Anisotropy ratio,

$$\frac{E_x}{E_y}$$

If isotropic, ANISO = 1.0

Col. 36-45: DELTA - Angle of friction, in degrees between pipe and soil.

Col. 46-55: FACTOR - Conversion factor for σ_{oct} and tangent modulus, e.g. σ_{oct} = FACTOR * σ_{oct} and E_t = FACTOR x E_t.
If no conversion is required, FACTOR = 1.0

(b) XP(I)

FORMAT (8F10.0)

σ_{oct} values of nonlinear property cards, total number values = NP, up to 8 values per card.

Col. 1-10: NP(1) - 1st value of σ_{oct}.

Col. 11-20: NP(2) - 2nd value of σ_{oct}.
and so on.

(c) EP, PSNR

FORMAT (2F10.0)

These set of cards define the nonlinear material properties. Two sets of curves are required (1) for tangent modulus, E_t vs σ_{oct} for stress-ratio ranging from 0.0 to 1.0 and (2) same type of curves for tangent Poisson's ratio ν_t, each curve of E_t and ν_t is defined by NP number of points, so total number of cards = NP x 11.

Col. 1-10: EP - Tangent modulus value.

Col. 11-20: PSNR - Tangent Poisson's ratio value

Note: (1) First NP cards should read the values of E_t and ν_t for increasing values of σ_{oct} starting from σ_{oct} = 0.
(2) The first set of NP cards are for curve of stress ratio = 1. The second set of NP cards will stand for stress ratio = 0.9 and so on. The last set of NP cards will read values of E_t and ν_t for stress-ratio = 0.0.

(3) A set of cards which includes cards from (a) to (c), define a complete set of Type 6 cards. As no material number has been attached to it, the sequence in which the set of cards are placed, will define the soil type. For example, the first complete set of type 6 cards (which includes (a), (b) and (c) type cards) will automatically be defined for Type 1 (NTYPE in 3rd Card Type) soil and the second set of type 2 and so on.

<u>7th Card Type</u>: NREAD, FACTOR

FORMAT (I5, F10.0)

Col. 1-5: NREAD - Total number of node point data cards. The triangular finite elements have six nodes i.e. three corner nodes and three mid-side nodes. Co-ordinates of mid-side nodes are calculated by the program. Except for defining boundary conditions, these nodes need not be defined.

Col. 6-15: FACTOR - Conversion factor for node point coordinates such as Ft. to Meter. If no conversion is required, FACTOR = 1.0.

<u>8th Card Type</u>: NN, NCODE, X, Y

FORMAT (I5, 5X, I2, F10.0, 20X, F10.0)

Nodal cards, one card per nodal point, total number = NREAD.

Col. 1-5: NN - Node point number.

Col. 11-12: NCODE - Node point boundary condition. The following table describes values of NCODE to represent desired boundary condition.

NCODE	Node Type	Boundary Condition
0	2 degrees of freedom in X and Y directions	Free X and Y directions.
1		Fixed in X and free in Y directions.
2		Free in X and fixed in Y directions.
3		Fixed in both X and Y directions.
4	3 degrees of freedom in X, Y direction and in rotation (θ)	Free in X, Y and θ directions.
5		Fixed in X, free in Y and θ directions.
6		Fixed in Y, free in X and θ directions.
7		Fixed in θ, free in X and Y directions.
8		Fixed in X and θ, free in Y directions.
9		Fixed in Y and θ, free in X directions.

Col. 13-22: X - X co-ordinate of node point.

Y - Y co-ordinate of node point.

Nodal data cards may be placed in any order.

9th Card Type: IEL, IX

FORMAT (I3, 2X, 2(3I5, 5X), 4X, I1, 3X, I2)

Element cards, one card per element total number = NELEMNT Cards.

Col. 1-3: IEL - Element number.

Col. 6-10: IX(1) - Node number 1 of element no. IEL.

Col. 11-15: IX(2) - Node number 2 of element no. IEL.

Col. 16-20: IX(3) - Node number 3 of element no. IEL.

Col. 26-30: IX(4) - Node number 4 of element no. IEL.

Col. 31-35: IX(5) - Node number 5 of element no. IEL.

Col. 36-40: IX(6) - Node number 6 of element no. IEL.

Col. 50: IX(7) - Element type identification number.

Col. 54-55: IX(8) - Material type identification number.

Element types 1, 2, 3 and corresponding node numbers are shown in Figure 1 to Figure 3. This numbering scheme has been used in the program. For type 4 and type 5 elements see Figures 4 and 5.

The element data cards can be placed in any order.

<u>10th Card Type</u>: ZAI(1), ZAI(2), ZAI(3)

FORMAT (3F10.0)

One card defining the point where stress of triangular element is required. (See Figure 12 for definition of ZAI).

Col. 1-10: ZAI(1) -

Col. 11-20: ZAI(2) - Area coordinates.

Col. 21-30: ZAI(3) -

<u>11th Card Type</u>: XCEN, YCEN, RADIUS, EI

FORMAT (3F10.0, E10.0)

One card defines center of circular pipe, radius and stiffness.

Col. 1-10: XCEN - X-coordinate of center of pipe.

YCEN - Y-coordinate of center of pipe.

RADIUS - Radius of pipe.

EI - Stiffness of pipe, Young's modulus times moment of inertia of pipe cross section per unit length.

<u>12th Card Type</u>: This card is required only when a plot of finite element mesh of the problem is required, which is specified in 1st Card Type. If Col. 79-80 is zero or blank no plot is generated. If any number is punched in Col. 79-80, a plot will be generated and in that case only this card is required.

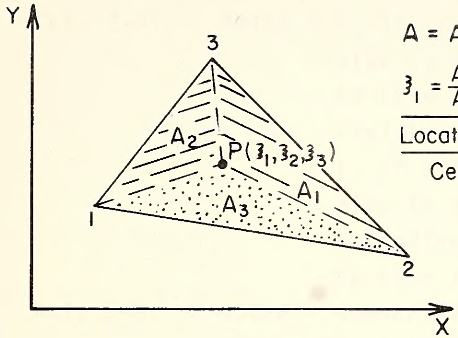
XMAX, YMAX, YCEN

FORMAT (3F10.0)

Col. 1-10: XMAX - Maximum size of mesh in X-direction.

YMAX - Maximum size of mesh in Y-direction.

YCEN - Y-coordinate of center of pipe.



$$A = A_1 + A_2 + A_3$$

$$\xi_1 = \frac{A_1}{A}, \ \xi_2 = \frac{A_2}{A}, \ \xi_3 = \frac{A_3}{A}$$

Location, P	ξ_1	ξ_2	ξ_3
Centroid	1/3	1/3	1/3
A+1	1	0	0
A+2	0	1	0
A+3	0	0	1

Figure 12. Definition of Area Co-ordinates

<u>13th Card Type</u>: NOTENSN, INTER, NSTEP, H1, H2, KN, KS, NITER
FORMAT (3I2, 4X, 2F10.0, 2E10.0, I5)
This card contains very important information regarding type of analysis required. Total number of cards = LAYERS (as specified in Col. 19-20 of 2nd Card Type).

Col. 1-2: NOTENSN = 0, if soil is allowed to take tension.
= 1, if 'no-tension' in soil analysis is required.

Col. 3-4: INTER = 0, if 'no-interaction' between soil and pipe is required in type II elements. In this case, pipe and soil are rigidly connected.
= 1, if interaction between soil and pipe is desired which will permit slip depending upon the state of stresses in type II elements.

Col. 5-6: NSTEP - Number of equal increments of application of gravity load in a particular layer.

Col. 11-20: H1 - Starting height of a construction layer being analyzed.

Col. 21-30: H2 - Finish height of a construction layer being analyzed.

Col. 31-40: KN - Normal stiffness of pipe-soil interaction element (type II) before failure in tension. In case tension develop in an interaction element, the program will modify value of this stiffness.

Col. 41-50: KS - Shear-stiffness of pipe-soil interaction element (type II) before failure. If INTER = 0, value of KN and KS are kept unchanged in all interaction elements throughout the analysis, which simulates rigid connection if values of KN and KS are considerably high. If INTER > 0 values of KN and KS are modified based on failure conditions specified for an interaction element.

Col. 51-55: NITER - Maximum number of iterations specified for 'no-tension' analysis. If the solution does not converge after specified number (= NITER) of iterations, farther execution will be stopped and a message will be printed.

<u>Input Data Cards for Program 'PROPRTY'</u>

In program FINLIN, an octahedral stress-strain system and tangent modulus and tangent Poisson's ratio values have been used for representing nonlinear soil properties. Required data for nonlinear properties in FINLIN (data card type 6), is difficult to get from conventional triaxial or plane strain tests.

In program FINLIN, some routine calculations and interpolations have to be performed to prepare data for card type 6. The program PROPRTY has been written to aid in generation of the data required for interpolation. This program accepts actual test data and interpolates using spline function, prints values of octahedral stress, strain, stress-ratio, tangent modulus, tangent Poisson's ratio and similar informations.

<u>1st Card Type</u>: TITL

FORMAT (10A8)

Col. 2-80: TITL - Alphanumeric identification of soil and test; one card.

<u>2nd Card Type</u>: TEST, NOCURVS, RF

FORMAT (A6, I3, F10.0)

One card defines type of test performed and other information.

Col. 1-6: TEST = PLSTRS for plane-stress test and
= PLSTRN for plane-strain test performed on the sample.

Col. 7-9: NOCURVS - Number of confining pressures used in the test; should at least be three.

Col. 10-19: RF - Ratio of τ_{oct}/σ_{oct} at failure.

<u>3rd Card Type</u>: SIGMA3, NP

FORMAT (5X, F10.0, I5)

One card specifies value of confining pressure and number of points for this test with given confining pressure.

Col. 6-15: SIGMA3 - Confining pressure.

Col. 16-20: NP - Number of points on a curve of given σ_3.

<u>4th Card Type</u>: XP, YP, VP

FORMAT (3F5.0)

One card per point, total number = NP; containing information about stress-strain and volume change data for a given σ_3.

Col. 1-5: XP - Axial strain in percent (ε_1 %).

Col. 6-10: YP - Deviator stress $(\sigma_1 - \sigma_3)$ corresponding to axial strain, XP.

Col. 11-15: VP - Volume strain in percent (ε_v %).

Note: 3rd and 4th card types are repeated NOCURVS times.

APPENDIX - I

FINLIN
PROGRAM LISTING

```
      OVERLAY(ROY,0,0)                                                  FLN
C                               OVERLAY(ROY,0,0)                        FLN
      PROGRAM FINLIN (INPUT,OUTPUT,TAPE2,TAPE9,TAPE5=INPUT,TAPE6=OUTPUT,FLN
     1TAPE1,TAPE3,TAPE4,TAPE7,TAPE10,PLOT)                              FLN
C                                                                       FLN
C       THIS FINITE ELEMENT PROGRAM HAS BEEN WRITTEN FOR PH.D. THESIS   FLN
C               PREDICTING PERFORMANCE OF PIPE CULVERTS IN SOIL         FLN
C                 BY M. B. ROY, GRADUATE RESEARCH ASSISTANT             FLN
C           SCHOOL OF CIVIL ENGINEERING, GEOTECHNICAL ENGINEERING,      FLN
C              PURDUE UNIVERSITY, WEST LAFAYETTE, INDIANA - 47907       FLN
C                          DATE DECEMBER, 1975.                         FLN
C                                                                       FLN
      COMMON NNODES,NELEMNT,NDOF,MBAND,ND,NT3,ISTOP,NCYCLE,LAYERS,ISTEP,FLN
     1NSTEP,NT12,ETA,NT1,NT2,MOTENSN,IFLAG,NSIZE,NCODE(550),X(550),Y(550FLN
     2),JNDX(51),ANLSIS,IX(8,250),AREAA(250),INDX(250),INDEX(250),GAMA(2FLN
     35),ZAI(3)                                                         FLN
      COMMON /1/ E,NUE,RADIUS,XCEN,YCEN,EI,KN,KS,H1,H2,INTER            FLN
      COMMON /2/ D                                                      FLN
      COMMON /3/ R                                                      FLN
      COMMON /4/ MODU(26),DELTA                                         FLN
      COMMON /5/ Q(1100),LIST(1101)                                     FLN
      DIMENSION TITLE(13), R(1100), TEMP(17)                            FLN
      DIMENSION E(10), NUE(10), D(10,10)                                FLN
      REAL NUE                                                          FLN
      REAL KN,KS                                                        FLN
C                                                                       FLN
C      THIS IS MAIN OVERLAY, WHICH DIRECTS EXECUTION OF OTHER OVERLAYS FLN
C             DEPENDING UPON TYPE OF ANALYSIS AND OTHER COMMANDS        FLN
C     FOLLOWING TWO STATEMENTS ARE LIBRARY ROUTINES FOR FOR BLOCK AND STFLN
C                                                                       FLN
      CALL FTNBIN (1,0)                                                 FLN
      CALL SETSTAK (0)                                                  FLN
C                                                                       FLN
C     MAX. SIZE OF SEMI-BAND WIDTHOF TOTAL STIFFNESS MATRIX INCLUDING DIFLN
C                                                                       FLN
      NSIZE=103                                                         FLN
      NSTEP=1                                                           FLN
      IFLAG=0                                                           FLN
      REWIND 2                                                          FLN
      READ (5,55) TITLE,MESH                                            FLN
      IF (EOF,5) 5,10                                                   FLN
    5 GO TO 40                                                          FLN
   10 CONTINUE                                                          FLN
      CALL ZERO (R,1100)                                                FLN
      WRITE (2) (R(I),I=1,1100)                                         FLN
      ISTEP=1                                                           FLN
      WRITE (6,60) TITLE                                                FLN
C                                                                       FLN
C                          CREATE MASS STORAGE FILES                    FLN
C                        INITIALISE MASS STORAGE UNITS                  FLN
C                                                                       FLN
      CALL OPENMS (1,INDX,250,0)                                        FLN
      CALL OPENMS (3,INDEX,250,0)                                       FLN
      CALL OPENMS (4,MODU,26,0)                                         FLN
      CALL OPENMS (7,JNDX,51,0)                                         FLN
      CALL OPENMS (10,LIST,1100,0)                                      FLN
      ND=NSIZE                                                          FLN
C                                                                       FLN
C         OVERLAY(1,0) READS PROBLEM GEOMETRY AND MATERIAL PROPERTIES   FLN
```

```
C
      CALL OVERLAY (3HROY,1,0,6HRECALL)
      IF (MESH.GT.0) CALL OVERLAY (3HROY,2,0,6HRECALL)
      CALL ZERO (TEMP,17)
      DO 15 I=1,NELEMNT
         CALL WRITMS (1,TEMP,17,I)
   15 CONTINUE
C
C           FOLLOWING PARAMETERS DESCRIBE TYPE OF ANALYSIS DESIRED
C       NOTENSN .GT. 0 = NO-TENSION ANALYSIS,, = 0 , NO CHECK FOR TENSION
C           INTER .GT. ZERO, = INTERACTION, INTER=0, NO INTERACTION
C                     NSTEP = NO. OF INCREMENTS PER LAYER
C                     H1 = STARTING HEIGHT OF THIS LAYER
C                      H2 = ENDING HEIGHT OF THIS LAYER
C                    KN = NORMAL STIFFNESS FOR INTERACTION
C                    KS = SHEAR STIFFNESS FOR INTERACTION
C         NITER = NO. OF ITERATION SPECIFIED FOR NO-TENSION ANALYSIS
C
   20 READ (5,65) NOTENSN,INTER,NSTEP,H1,H2,KN,KS,NITER
      NCYCLE=1
      WRITE (6,70) INTER,NSTEP,H1,H2,NOTENSN
   25 ND=NSIZE
      NCOUNT=0
C
C              FORM STRUCTURAL STIFFNESS MATRIX AND LOAD VECTOR
C
      CALL OVERLAY (3HROY,3,0,6HRECALL)
      WRITE (6,75)
C
C                        MODIFY FOR BOUNDARY CONDITIONS
C                            SOLVE FOR DISPLACEMENT
C
      CALL OVERLAY (3HROY,6,0,6HRECALL)
C
C                FIND STRESSES AND STRAINS, AND PRINT RESULTS
C
   30 CALL OVERLAY (3HROY,4,0,6HRECALL)
      CALL OVERLAY (3HROY,5,0,6HRECALL)
      IF (IFLAG.LE.0) GO TO 35
C
C                            RECYCLE FOR NO-TENSION
C
      NCOUNT=NCOUNT+1
      IF (NCOUNT.GT.NITER) GO TO 50
      CALL OVERLAY (3HROY,6,0,6HRECALL)
      GO TO 30
C
C           RECYCLE PROGRAM FOR INCREMENTAL OR NON-LINEAR ANALYSIS
C
   35 NCYCLE=NCYCLE+1
      IFLAG=0
      IF (NCYCLE.GT.NSTEP) GO TO 40
      GO TO 25
   40 ISTEP=ISTEP+1
      IF (ISTEP.GT.LAYERS) GO TO 45
      GO TO 20
   45 WRITE (6,80)
C
C                         CLOSE ALL MASS STORAGE UNITS
C
```

```
      CALL CLOSEMS (1)
      CALL CLOSEMS (3)
      CALL CLOSEMS (4)
      CALL CLOSEMS (7)
      CALL CLOSEMS (10)
C
      STOP
   50 WRITE (6,85) NCOUNT
      STOP
C
   55 FORMAT (13A6,I2)
   60 FORMAT (20X,13A6)
   65 FORMAT (3I2,4X,2F10.0,2E10.0,I5)
   70 FORMAT (/10X, 17HLAYER INFORMATION,/10X, 14HINTERACTION = ,I5/10X,
     1 35HNO. OF INCREMENTS FOR THIS LAYER = ,I5/10X, 32HSTARTING HEIGHT
     2 OF THIS LAYER = ,F10.2/10X, 33HFINISHING HEIGHT OF THIS LAYER = ,
     3F10.2/10X, 14H NO-TENSION = ,I5/10X, 30HNO. OF ITERATIONS SPECIFIE
     4D = ,I5//)
   75 FORMAT (10X, 23HOVERLAY (3,0) COMPLETED)
   80 FORMAT (////5X, 15H END OF PROBLEM)
   85 FORMAT (//,10X, 45HNO-TENSION ITERATION DOES NOT CONVERGE AFTER ,I
     15, 13H    ITERATIONS,/,10X, 20HEXECUTION TERMINATED)
C
      END
      SUBROUTINE ZERO (A,N)
C
C     ***************************************************************************
C                    THIS SUBROUTINE GENERATES A NULL VECTOR
C     ***************************************************************************
C
      DIMENSION A(N)
      DO 5 I=1,N
         A(I)=0.0
    5 CONTINUE
      RETURN
C
      END
      SUBROUTINE NULLMAT (A,M,N)
C
C          THIS ROUTINE GENERATES A NULL MATRIX ≠ A ≠ OF SIZE (M X N)
C
      DIMENSION A(M,N)
      DO 5 I=1,M
      DO 5 J=1,N
         A(I,J)=0.0
    5 CONTINUE
      RETURN
C
      END
      OVERLAY(ROY,1,0)
C                                 OVERLAY(ROY,1,0)
      PROGRAM INPUT1
      COMMON NNODES,NELEMNT,NDOF,MBAND,ND,NT3,ISTOP,NCYCLE,LAYERS,ISTEP,
     1NSTEP,NT12,ETA,NT1,NT2,NOTENSN,IFLAG,NSIZE,NCODE(550),X(550),Y(550
     2),JNDX(51),ANLSIS,IX(8,250),AREAA(250),INDX(250),INDEX(250),GAMA(2
     35),ZAI(3)
      COMMON /1/ E,NUE,RADIUS,XCEN,YCEN,EI,KN,KS,H1,H2,INTER
      COMMON /2/ D
      COMMON /4/ MODU(26),DELTA
      DIMENSION E(10), NUE(10), D(10,10)
```

```
      DIMENSION TYPE(10)
      DIMENSION NL(4)
      DATA MAXNP,MAXEL,MAXMAT/550,250,25/
      DATA PLSTRS,PLSTRN/6HPLSTRS,6HPLSTRN/
      REAL KZERO
      REAL KN,KS
      REAL NUE
      NT3=0
      NDOF=0
      MBAND=0
      NT1=0
      NT2=0
      NT12=0
C
C                 READ PROBLEM STATEMENT AND OTHER PARAMETERS
C
      READ (5,75) NNODES,NELEMNT,MATRIAL,NPRSR,LAYERS,ISTOP
      IF (NNODES.LE.MAXNP) GO TO 5
      WRITE (6,80) NNODES
    5 IF (NELEMNT.LE.MAXEL) GO TO 10
      WRITE (6,85) NELEMNT
C
C     *******************************************************************
C                        PRINT LAYOUT OF THE PROBLEM
C     *******************************************************************
C
   10 WRITE (6,90) NNODES,NELEMNT,MATRIAL,NPRSR,LAYERS
      IF (MATRIAL.LE.MAXMAT) GO TO 15
      WRITE (6,95) MATRIAL
   15 CONTINUE
C
C      ******************************************************************
C                READ MATERIAL TYPES, PROPERTIES AND DESCRIPTION
C      ******************************************************************
C
      WRITE (6,105)
      CALL ZERO (GAMA,25)
      DO 20 I=1,MATRIAL
         READ (5,100) NTYPE,GAMA(NTYPE),TYPE
         WRITE (6,110) NTYPE,GAMA(NTYPE),TYPE
   20 CONTINUE
      READ (5,115) ANLSIS,DELTA
      DO 25 I=1,MATRIAL
         CALL INPUT2 (I)
         IF (ANLSIS.EQ.PLSTRS) CALL PLNSTRS (I)
         IF (ANLSIS.EQ.PLSTRN) CALL PARAMTR (I)
C
C              IF ISTOP .GT. ZERO, NON-LINEAR PROPERTY SPECIFIED
C
         IF (ISTOP.GT.0) CALL SPLINE (I)
   25 CONTINUE
C
C     *******************************************************************
C
      CALL ZERO (X,NNODES)
      CALL ZERO (Y,NNODES)
      CALL ZERO (NCODE,550)
C
C            READ NODAL POINT DATA, CODE, COORDINATES, LOADS, ETC.
C               NREAD = TOTAL NO. OF NODAL DATA CARDS SUPPLIED
```

```
C         FACTOR = UNIT CONVERSION FACTOR FOR NODAL COORDINATE DATA
C
      READ (5,120) NREAD,FACTOR
      DO 30 I=1,NREAD
         READ (5,125) NN,NCODE(NN),X(NN),Y(NN)
         IF (NCODE(NN).GT.3) NT3=NT3+1
         X(NN)=X(NN)*FACTOR
         Y(NN)=Y(NN)*FACTOR
   30 CONTINUE
C
C     ***********************************************************************
C                          READ ELEMENT DATA, TYPE, MATERIAL
C
      CALL NULLMAT (IX,8,NELEMNT)
      CALL ZERO (AREAA,NELEMNT)
      DO 35 I=1,NELEMNT
         READ (5,130) IEL,(IX(J,IEL),J=1,8)
         NT=IX(7,IEL)
         IF (NT.EQ.1) NT1=NT1+1
         IF (NT.EQ.2) NT2=NT2+1
   35 CONTINUE
      NT12=NT1+NT2
C
C                          WRITE NODAL POINT INFORMATIONS
C
      WRITE (6,135)
      DO 40 I=1,NNODES
         WRITE (6,140) I,NCODE(I),X(I),Y(I)
   40 CONTINUE
C
C                            WRITE ELEMENT INFORMATIONS
C
      WRITE (6,145)
      DO 45 I=1,NELEMNT
         WRITE (6,150) I,(IX(J,I),J=1,8)
   45 CONTINUE
      READ (5,160) ZAI(1),ZAI(2),ZAI(3)
      WRITE (6,155) ZAI(1),ZAI(2),ZAI(3)
C
C                                  DETERMINE BAND WIDTH
C
C        DETERMINE NODAL FORCES DUE TO SELF WEIGHT, FOR THIS PURPOSE
C
      WRITE (6,165)
      DO 65 II=1,NELEMNT
         NT=IX(7,II)
         GO TO (50,55,60,60,60), NT
   50    I1=IX(1,II)
         I2=IX(2,II)
         IDF=I1-I2
         NBAND=(IABS(IDF)+1)*3
         IF (NBAND.GT.MBAND) MBAND=NBAND
         GO TO 65
   55    I1=IX(1,II)
         I2=IX(2,II)
         I3=IX(3,II)
         I4=IX(4,II)
         NL(1)=3*I1-2
         NL(2)=3*I2-2
         NL(3)=2*I3-1+NT3
```

```
      NL(4)=2*I4-1+NT3
      MAX=MAX0(NL(1),NL(2),NL(3),NL(4))
      MIN=MIN0(NL(1),NL(2),NL(3),NL(4))
      NBAND=MAX-MIN+2
      IF (NBAND.GT.MBAND) MBAND=NBAND
      GO TO 65
   60 CALL AREAS (II,NT)
   65 CONTINUE
      NDOF=2*NNODES+NT3
C
C                          PRINT SIZE OF THE PROBLEM
C
      WRITE (6,170) NDOF,MBAND
C
      IF (NT3.LE.0) GO TO 70
C
C                     READ CENTER, RADIUS, AND EI OF PIPE
C
      READ (5,175) XCEN,YCEN,RADIUS,EI
      WRITE (6,180) XCEN,YCEN,RADIUS,EI
C
C                    DETERMINE ETA FOR TRIANGULAR ELEMENTS
C
      NT12P1=NT12+1
      IF (NT12P1.GT.NELEMNT) GO TO 70
      I1=IX(1,NT12P1)
      I2=IX(2,NT12P1)
      X1=X(I1)
      Y1=Y(I1)
      X2=X(I2)
      Y2=Y(I2)
      SPAN=SQRT((X1-X2)**2+(Y1-Y2)**2)
      SN=0.5*SPAN/RADIUS
      TH=ASIN(SN)
      THETA=2.*TH
      R2=RADIUS**2
      A1=0.5*THETA*R2
      A2=0.5*R2*SIN(THETA)
      A4=A1-A2
      ETA=A4/AREAA(NT12P1)
      ETA=0.0
C
   70 CONTINUE
      WRITE (6,185)
      RETURN
C
   75 FORMAT (2I5,2I3,2X,I2,2X,I2,1X,I1,1X,I2)
   80 FORMAT (5X, 42HNO. OF NODES EXCEEDS LIMIT(=900), NNODES= ,I5)
   85 FORMAT (5X, 46HNO. OF ELEMENTS EXCEEDS LIMIT(=500), NELEMNT= ,I5)
   90 FORMAT (10X, 19HPROBLEM DESCRIPTION,//5X, 19HNO. OF NODE POINTS=,I
     15/5X, 16HNO. OF ELEMENTS=,I5/5X, 17HNO. OF MATERIALS=,I5/5X, 26HNO
     2. OF BOUNDARY PRESSURES=,I5/5X, 29HNO. OF CONSTRUCTION LAYERS = ,I
     35//)
   95 FORMAT (5X, 47HNO. OF MATERIALS EXCEEDS LIMIT (= 25), MATRIAL=,I5)
  100 FORMAT (I5,F10.0,10A6)
  105 FORMAT (///10X, 21HMATERIAL DESCRIPTIONS,//5X,  3HNO.,5X, 11HUNIT
     1WEIGHT,10X, 11HDESCRIPTION,/)
  110 FORMAT (5X,I2,8X,F8.4,10X,10A6)
  115 FORMAT (A6,F5.0)
  120 FORMAT (I5,F10.0)
```

```
  125 FORMAT (I5,5X,I2,F10.0,20X,F10.0)
  130 FORMAT (I3,2X,2(3I5,5X),4X,I1,3X,I2,2F5.0,5X,2F5.0)
  135 FORMAT (///10X, 16HNODAL POINT DATA,//5X,  8HNODE NO.,2X,  9HDIREC
     1TION,3X, 10HFIXED/FREE,5X, 11HCO-ORDINATE,5X, 10HNODAL LOAD,2X,  9
     2HDIRECTION,3X, 10HFIXED/FREE,5X, 11HCO-ORDINATE,5X, 10HNODAL LOAD,
     3/)
  140 FORMAT (7X,I4,8X,  1HX,10X,I1,11X,F8.2,24X,  1HY,22X,F8.2)
  145 FORMAT (///10X, 18HELEMENT DATA INPUT,//1X, 11HELEMENT NO.,2X, 59H
     1 -P-  -Q-  -R-  -S-  -PQ-  -QR-  -RP-  -RS-  TYPE  MATERIAL,//)
  150 FORMAT (2X,I5,6X,3(I3,2X),5X,3(I4,2X),6X,2X,I2,5X,I3,4X,F6.1,2X,F6
     1.2,1X,F4.0,1X,F4.0)
  155 FORMAT (10X,  9HZAI(1) = ,F5.2,5X,  9HZAI(2) = ,F5.2,5X,  9HZAI(3)
     1 = ,F5.2/)
  160 FORMAT (3F10.0)
  165 FORMAT (/10X, 20HELEMENT INFORMATIONS,/1X,  7HELE.NO.,9X,  6HZAI(1
     1),9X,  6HZAI(2),9X,  6HZAI(3),10X,  4HAREA,5X, 24HX-COORDINATE,Y-C
     2ORDINATE,/)
  170 FORMAT (///10X, 12HPROBLEM SIZE,//25X, 27HNO. OF DEGREES OF FREEDO
     1M =,I5/35X, 17HSEMI-BAND-WIDTH =,I5/)
  175 FORMAT (3F10.0,E10.0)
  180 FORMAT (/10X, 19HPIPE SPECIFICATIONS,/10X, 11HX-CENTER = ,F6.2,3X,
     1 11HY-CENTER = ,F6.2,/10X, 14HPIPE RADIUS = ,F6.2,/10X, 15HPIPE ST
     2IFFNESS ,E12.2/)
  185 FORMAT (1X, 23HOVERLAY (1,0) COMPLETED)
C
      END
      SUBROUTINE INPUT2 (II)
      COMMON /1/ E,NUE,RADIUS,XCEN,YCEN,EI,KN,KS,H1,H2,INTER
      DIMENSION E(10), NUE(10)
      REAL KN,KS
C
C     ****************************************************************
C       THIS ROUTINE READS DATA AND NECESSARY VARIATIONAL PARAMETERS FOR
C              MATERIAL PROPERTIES AND CONSTITUTIVE RELATIONSHIP.
C        THIS ROUTINE IS DIRECTLY CONNECTED TO ≠RARAMTR ≠ THROUGH COMMON
C
      REAL NUE
C
C                         READ E , NUE FOR FIXED PROPERTY
C
      READ (5,5) E(II)
      READ (5,10) NUE(II)
      WRITE (6,15) II,E(II),NUE(II)
      RETURN
C
    5 FORMAT (10E8.0)
   10 FORMAT (F5.0)
   15 FORMAT (////10X, 27HPROPERTIES FOR MATERIAL NO ,I5,//5X, 15HYOUNGS
     1 MODULUS=,1X,E11.4/5X, 15HPOISSONS RATIO=,F5.2)
C
      END
      SUBROUTINE PLNSTRS (K)
      COMMON /1/ E,NUE,RADIUS,XCEN,YCEN,EI,KN,KS,H1,H2,INTER
      COMMON /2/ D
      REAL NUE,NUEK
      REAL KN,KS
      DIMENSION D(10,10), E(10), NUE(10)
C
C                         THIS ROUTINE IS FOR PLANE STRESS
C
```

```
      NUEK=NUE(K)
      C=E(K)/(1.-NUEK*NUEK)
      D(K,8)=0.0
      D(K,7)=D(K,8)
      D(K,6)=D(K,7)
      D(K,3)=D(K,6)
      D(K,5)=C
      D(K,1)=D(K,5)
      D(K,4)=C*NUEK
      D(K,2)=D(K,4)
      D(K,9)=0.5*C*(1.-NUEK)
      D(K,10)=NUEK
      RETURN
C
      END
      SUBROUTINE PARAMTR (K)
      COMMON /1/ E,NUE,RADIUS,XCEN,YCEN,EI,KN,KS,H1,H2,INTER
      COMMON /2/ D
      REAL KN,KS
C
C                    THIS ROUTINE IS FOR PLANE STRAIN
C
      REAL NUE,NUEK
      DIMENSION D(10,10), E(10), NUE(10)
      NUEK=NUE(K)
      C=(E(K)*(1.-NUEK))/((1.+NUEK)*(1.-2.*NUEK))
      D(K,5)=C
      D(K,1)=D(K,5)
      D(K,4)=C*NUEK/(1.-NUEK)
      D(K,2)=D(K,4)
      D(K,9)=C*(1.-2.*NUEK)/(2.*(1.-NUEK))
      D(K,8)=0.0
      D(K,7)=D(K,8)
      D(K,6)=D(K,7)
      D(K,3)=D(K,6)
      D(K,10)=NUEK
      RETURN
C
      END
      SUBROUTINE AREAS (MN,NT)
      COMMON NNODES,NELEMNT,NDUF,MBAND,ND,NT3,ISTOP,NCYCLE,LAYERS,ISTEP,
     1NSTEP,NT12,ETA,NT1,NT2,NOTENSN,IFLAG,NSIZE,NCODE(550),X(550),Y(550
     2),JNDX(51),ANLSIS,IX(8,250),AREAA(250),INDX(250),INDEX(250),GAMA(2
     35),ZAI(3)
      DIMENSION LM(6)
C
C     ******************************************************************
C          THIS ROUTINE CALCULATES ELEMENT AREA, AND SEMI BAND WIDTH
C
      AREA=0.0
      I=IX(1,MN)
      J=IX(2,MN)
      K=IX(3,MN)
      MK=NT-2
      GO TO (5,15,20), MK
    5 DO 10 N=1,6
   10 LM(N)=2*IX(N,MN)+NT3-1
      GO TO 30
   15 LM(1)=3*IX(1,MN)-2
      LM(2)=3*IX(2,MN)-2
```

```
      LM(3)=2*IX(3,MN)-1+NT3
      LM(4)=3*IX(4,MN)-2
      LM(5)=2*IX(5,MN)-1+NT3
      LM(6)=2*IX(6,MN)-1+NT3
      GO TO 30
   20 DO 25 N=1,6
   25 LM(N)=2*IX(N,MN)-1+NT3
      LM(3)=3*IX(3,MN)-2
   30 AREA=ABS(.5*(X(J)*Y(K)-Y(J)*X(K)+X(I)*(Y(J)-Y(K))+Y(I)*(X(K)-X(J))
     1))
      MAX=MAX0(LM(1),LM(2),LM(3),LM(4),LM(5),LM(6))
      MIN=MIN0(LM(1),LM(2),LM(3),LM(4),LM(5),LM(6))
      NWIDTH=MAX-MIN+2
      IF (NWIDTH.GT.MBAND) MBAND=NWIDTH
      IF (NWIDTH.GT.ND) WRITE (6,40) NWIDTH,MN
      IF (AREA.GT.0.0) GO TO 35
      WRITE (6,45) MN
      RETURN
C
   35 XX=ZAI(1)*X(I)+ZAI(2)*X(J)+ZAI(3)*X(K)
      YY=ZAI(1)*Y(I)+ZAI(2)*Y(J)+ZAI(3)*Y(K)
      AREAA(MN)=AREA
      WRITE (6,50) MN,(ZAI(I),I=1,3),AREA,XX,YY
      RETURN
C
   40 FORMAT (3X, 32HBANDWIDTH EXCEEDS LIMIT, MBAND= ,I5,5X, 17HIN ELEME
     1NT NO. = ,I5)
   45 FORMAT (5X, 35HNEGATIVE OR ZERO AREA, ELEMENT NO.=,I5)
   50 FORMAT (2X,I5,5X,4E14.2,2F12.2)
C
      END
      SUBROUTINE SPLINE (IMAT)
      COMMON /4/ MODU(26),DELTA
      DIMENSION XP(7), EP(11,7), SEP(11,7), PSNR(11,7), SPSNR(11,7)
      DIMENSION PROP(319), SRO(11)
C
C        SPLINE FITTING FOR TAN. MOD, TAN. POISSSON RATIO VS, SIGMAOCTA
C                          NP= NO. OF DATA POINTS FOR STRESS
C                       PHI= FRICTION ANGLE FOR SOIL (DEGREES)
C              PSI= RATIO OF (SIGMA2)/(SIGMA1+SIGMA3) FOR PLANE STRAIN
C                   ANISO= DEGREE OF ANISOTROPY OF SOIL STRENGTH
C                          FOR ISOTROPIC MATERIAL ANISO=1.
C           DELTA= ANGLE OF FRICTION ( DEGREES ) BETWEEN SOIL AND PIPE
C                    FACTOR= STREHGTH PROPERTY CONVERSION FACTOR
C
      READ (5,55) NP,PSI,PHI,ANISO,DELTA,FACTOR
      IF (ANISO.LE.0.0) ANISO=1.0
      FRATIO=FAILURE(PSI,PHI)
C
C                           XP= OCTAHEDRAL NORMAL STRESS VECTOR
C                                  EP= TANGENT MODULUS
C                              PSNR= TANGENT POISSONS RATIO
C
      READ (5,65) (XP(I),I=1,NP)
      DO 5 I=1,NP
    5 XP(I)=FACTOR*XP(I)
      DO 15 NO=1,11
         READ (5,70) (EP(NO,J),PSNR(NO,J),J=1,NP)
         SRO(NO)=(11.-FLOAT(NO))/10.
         DO 10 J=1,NP
```

```
 10     EP(NO,J)=FACTOR*EP(NO,J)
C
C         CUBIC SPLINE FITTING FOR TANGENT MODULUS AND POISSONS RATIO
C
        CALL SPLFIT (NP,NO,XP,EP,SEP)
        CALL SPLFIT (NP,NO,XP,PSNR,SPSNR)
 15 CONTINUE
    WRITE (6,75) IMAT
    WRITE (6,60) PHI,PSI,FRATIO,DELTA,ANISO
    WRITE (6,80) (XP(I),I=1,NP)
    WRITE (6,85) (SRO(I),I=1,11)
    DO 20 I=1,NP
 20 WRITE (6,90) (EP(J,I),J=1,11)
    WRITE (6,95) (SRO(I),I=1,11)
    DO 25 I=1,NP
 25 WRITE (6,90) (PSNR(J,I),J=1,11)
C
C              STORE MATERIAL PROPERTIES ON MASS STORAGE UNIT
C
    PROP(1)=ANISO
    PROP(2)=DELTA
    PROP(3)=FRATIO
    MO=3
    DO 30 I=1,7
       MO=MO+1
 30 PROP(MO)=XP(I)
    DO 35 I=1,11
    DO 35 J=1,7
       MO=MO+1
 35 PROP(MO)=EP(I,J)
    DO 40 I=1,11
    DO 40 J=1,7
       MO=MO+1
 40 PROP(MO)=SEP(I,J)
    DO 45 I=1,11
    DO 45 J=1,7
       MO=MO+1
 45 PROP(MO)=PSNR(I,J)
    DO 50 I=1,11
    DO 50 J=1,7
       MO=MO+1
 50 PROP(MO)=SPSNR(I,J)
    PROP(319)=FLOAT(NP)
    CALL WRITMS (4,PROP,319,IMAT)
    RETURN
C
 55 FORMAT (I5,5F10.0)
 60 FORMAT (1X, 24HFAILURE RATIO FOR PHI = ,F6.2, 12H  AND PSI = ,F5.2
   1,  8H  IS =  ,F6.3/10X,  8HDELTA = ,F6.2, 22H  ANISOTROPY FACTOR =
   2 ,F6.2/)
 65 FORMAT (8F10.0)
 70 FORMAT (2F10.0)
 75 FORMAT (/10X, 45HNON-LINEAR SOIL PROPERTIES FOR MATERIAL NO = ,I5)
 80 FORMAT (1X, 11HSIGMA(OCT)=,12E9.1)
 85 FORMAT (10X, 22HTANGENT MODULUS VALUES,/2X, 15HSTRESS RATIO = ,F3.
   11,10(6X,F3.1)/)
 90 FORMAT (15X,    11E9.2)
 95 FORMAT (//10X, 28HTANGENT POISSON RATIO VALUES,/2X, 15HSTRESS RATI
   1O = ,F3.1,10(6X,F3.1)/)
C
```

```
      END
      SUBROUTINE SPLFIT (NPN,NO,XP,YP,YDP)
      DIMENSION XP(7), YP(11,7), YDP(11,7), H(7), Y(7), AI(7), BI(7), CI
     1(7), DI(7), SDP(7)
C
C                           CUBIC SPLINE FITTING
C
      NP1=NPN-1
      DO 5 M=1,NP1
    5 H(M)=XP(M+1)-XP(M)
      SLOP1=FD(H(1),H(2),YP(NO,1),YP(NO,2),YP(NO,3))
      SLOPN=BD(H(NPN-2),H(NPN-1),YP(NO,NPN-2),YP(NO,NPN-1),YP(NO,NPN))
      DO 10 M=1,NPN
   10 Y(M)=YP(NO,M)
      CALL COFRIT (NPN,XP,Y,SLOP1,SLOPN,AI,BI,CI,DI)
      CALL TRIDGNL (NPN,AI,BI,CI,DI,SDP)
      DO 15 I=1,NPN
   15 YDP(NO,I)=SDP(I)
      RETURN
C
      END
      SUBROUTINE COFRIT (NPN,XP,YP,SLOP1,SLOPN,AI,BI,CI,DI)
      DIMENSION XP(7), YP(7), AI(7), BI(7), CI(7), DI(7)
C
C                      GENERATE SPLINE COEFFICIENTS
C
      AI(1)=0.0
      BI(1)=(XP(2)-XP(1))/3.
      CI(1)=BI(1)/2.
      DI(1)=(YP(2)-YP(1))/(XP(2)-XP(1))-SLOP1
      AI(NPN)=(XP(NPN)-XP(NPN-1))/6.
      BI(NPN)=AI(NPN)*2.
      CI(NPN)=0.0
      DI(NPN)=-(YP(NPN)-YP(NPN-1))/(XP(NPN)-XP(NPN-1))+SLOPN
      N1=NPN-1
      DO 5 I=2,N1
         AI(I)=(XP(I)-XP(I-1))/6.
         BI(I)=(XP(I+1)-XP(I-1))/3.
         CI(I)=(XP(I+1)-XP(I))/6.
         DI(I)=(YP(I+1)-YP(I))/(XP(I+1)-XP(I))-(YP(I)-YP(I-1))/(XP(I)-XP
     1   (I-1))
    5 CONTINUE
      RETURN
C
      END
      SUBROUTINE TRIDGNL (NPN,AI,BI,CI,DI,YDP)
      DIMENSION AI(7), BI(7), CI(7), DI(7), YDP(7), Q(10), U(10)
C
C                         SOLVE TRI-DIAGONAL MATRIX
C
      P=BI(1)
      Q(1)=-CI(1)/P
      U(1)=DI(1)/P
      DO 5 K=2,NPN
         P=AI(K)*Q(K-1)+BI(K)
         Q(K)=-CI(K)/P
         U(K)=(DI(K)-AI(K)*U(K-1))/P
    5 CONTINUE
      YDP(NPN)=U(NPN)
      N1=NPN-1
```

```
      DO 10 L=1,N1
         K=N1+1-L
         YDP(K)=Q(K)*YDP(K+1)+U(K)
   10 CONTINUE
      RETURN
C
      END
      FUNCTION FD(S1,S2,R1,R2,R3)
C
C                           FOREWARD DIFFERENCE FUNCTION
C
      IF (S1-S2) 5,10,5
    5 FD=(R2-R1)/S1
      RETURN
   10 FD=(-3.*R1+4.*R2-R3)/(2.*S1)
      RETURN
C
      END
      FUNCTION BD(S1,S2,R1,R2,R3)
C
C                           BACKWARD DIFFERENCE FUNCTION
C
      IF (S1-S2) 5,10,5
    5 BD=(R3-R2)/S2
      RETURN
   10 BD=(3.*R3-4.*R2+R1)/(2.*S1)
      RETURN
C
      END
      FUNCTION ORDINET(A,B,C,D,E,F,P)
      HJ=D-C
      B1=D-P
      B2=P-C
      A1=B1**3
      A2=B2**3
      T1=A1*A/(6.*HJ)
      T2=A2*B/(6.*HJ)
      T3=(E-A*HJ**2/6.)*(D-P)/HJ
      T4=(F-B*HJ**2/6.)*(P-C)/HJ
      ORDINET=T1+T2+T3+T4
      RETURN
C
      END
      FUNCTION FAILURE(PSI,PHII)
C
C                         DETERMINE FAILURE STRESS RATIO
C
      REAL NPHI
      PI=22./7.
      PHI=PHII*PI/180.
      SINPHI=SIN(PHI)
      NPHI=(1.+SINPHI)/(1.-SINPHI)
      T1=6.*NPHI/((NPHI+1.)**2)
      T2=2.*(PSI**2-PSI+1.)
      T3=SQRT(T2-T1)
      T4=T3/(1.+PSI)
      FAILURE=T4
      RETURN
C
      END
```

```
      OVERLAY(ROY,2,0)
C                                OVERLAY(ROY,2,0)
      PROGRAM LAYOUT
      COMMON NNODES,NELEMNT,NDOF,MBAND,ND,NT3,ISTOP,NCYCLE,LAYERS,ISTEP,
     1NSTEP,NT12,ETA,NT1,NT2,NOTENSN,IFLAG,NSIZE,NCODE(550),X(550),Y(550
     2),JNDX(51),ANLSIS,IX(8,250),AREAA(250),INDX(250),INDEX(250),GAMA(2
     35),ZAI(3)
      DIMENSION XPOLY(6), YPOLY(6)
      CALL PLOTS
      READ (5,25) XMAX,YMAX,YCEN
      SF=10./YMAX
      CALL FACTOR (SF)
      CALL PLOT (0.0,0.0,-3)
      YPOLY(4)=0.0
      XPOLY(4)=YPOLY(4)
      YPOLY(5)=1.0
      XPOLY(5)=YPOLY(5)
      CALL AXIS (0.0,0.0,2HX-,-2,XMAX,0.0,XPOLY(4),XPOLY(5),0)
      CALL AXIS (0.0,0.0,2HY-,2,YMAX,90.0,YPOLY(4),YPOLY(5),-1)
      CALL PLOT (0.0,YMAX,3)
      CALL PLOT (XMAX,YMAX,2)
      CALL PLOT (XMAX,0.0,2)
      DO 20 I=1,NELEMNT
         NT=IX(7,I)
         GO TO (5,10,15), NT
    5    I1=IX(2,I)
         I2=IX(1,I)
         XPOLY(1)=X(I1)
         YPOLY(1)=Y(I1)
         XPOLY(2)=X(I2)
         YPOLY(2)=Y(I2)
         YPOLY(3)=0.0
         XPOLY(3)=YPOLY(3)
         YPOLY(4)=1.0
         XPOLY(4)=YPOLY(4)
         CALL LINE (XPOLY,YPOLY,2,1,0,0)
         GO TO 20
   10    CONTINUE
   15    I1=IX(1,I)
         I2=IX(2,I)
         I3=IX(3,I)
         XPOLY(4)=X(I1)
         XPOLY(1)=XPOLY(4)
         YPOLY(4)=Y(I1)
         YPOLY(1)=YPOLY(4)
         XPOLY(2)=X(I2)
         YPOLY(2)=Y(I2)
         XPOLY(3)=X(I3)
         YPOLY(3)=Y(I3)
         YPOLY(5)=0.0
         XPOLY(5)=YPOLY(5)
         YPOLY(6)=1.0
         XPOLY(6)=YPOLY(6)
         CALL LINE (XPOLY,YPOLY,4,1,0,0)
         XC=(X(I1)+X(I2)+X(I3))/3.-0.2
         YC=(Y(I1)+Y(I2)+Y(I3))/3.
         CALL NUMBER (XC,YC,0.4,I,0.0,2HI3)
   20 CONTINUE
      XC=XMAX/2.-2.
      YC=YMAX+2.
```

```
      CALL SYMBOL (0.0,YCEN,0.3,11,0.0,-1)
      CALL SYMBOL (XC,YC,0.6,19HFINITE ELEMENT MESH,0.0,19)
      CALL PLOT (0.0,0.0,999)
      RETURN
C
   25 FORMAT (3F10.0)
C
      END
      OVERLAY(ROY,3,0)
C                                    OVERLAY(ROY,3,0)
      PROGRAM BLOCK
      COMMON NNODES,NELEMNT,NDOF,MBAND,ND,NT3,ISTOP,NCYCLE,LAYERS,ISTEP,
     1NSTEP,NT12,ETA,NT1,NT2,NOTENSN,IFLAG,NSIZE,NCODE(550),X(550),Y(550
     2),JNDX(51),ANLSIS,IX(8,250),AREAA(250),INDX(250),INDEX(250),GAMA(2
     35),ZAI(3)
      COMMON /1/ E,NUE,RADIUS,XCEN,YCEN,EI,KN,KS,H1,H2,INTER
      COMMON /3/ R
      COMMON /4/ MODU(26),DELTA
      COMMON /5/ Q(1100),LIST(1101)
      DIMENSION R(1100), TEMP(17), LM(6), SK(12,12)
      DIMENSION XP(7), EP(11,7), SEP(11,7), PSNR(11,7), SPSNR(11,7)
      DIMENSION E(10), NUE(10), PROP(319)
      DIMENSION A(103,103), ARRAY(103)
      DIMENSION SR(6,6), SKJT(8,8)
      DIMENSION TENP(17)
      REAL NUEX,NUEY
      REAL KN,KS
      REAL NUE
      PI=22./7.
      ANISO=1.0
      NDINC=1
      IF (NT3.LE.0) NDINC=0
      IF (NT3.LE.0) ND=ND-1
      NEND=3*NT3
      ND2=2*ND
      NDOF=2*NNODES+NT3
      CALL NULLMAT (A,NSIZE,NSIZE)
      CALL NULLMAT (SK,12,12)
      CALL ZERO (Q,NDOF)
C
C           FIND NODAL LOADS FOR ELEMENTS IN THIS PARTICULAR LAYER
C
      DO 40 II=1,NELEMNT
         NT=IX(7,II)
         GO TO (40,40,5,5,5), NT
    5    I1=IX(1,II)
         I2=IX(2,II)
         I3=IX(3,II)
         YC=(Y(I1)+Y(I2)+Y(I3))/3.
         IF (YC.GT.H2.OR.YC.LE.H1) GO TO 40
         MT=IX(8,II)
         WT=GAMA(MT)*AREAA(II)
         MK=MT-2
         GO TO (10,20,25), MK
   10    DO 15 I=1,6
   15    LM(I)=2*IX(I,II)+NT3
         GO TO 35
   20    LM(1)=3*IX(1,II)-1
         LM(2)=3*IX(2,II)-1
         LM(3)=2*IX(3,II)+NT3
```

```
        LM(4)=3*IX(4,II)-1
        LM(5)=2*IX(5,II)+NT3
        LM(6)=2*IX(6,II)+NT3
        GO TO 35
 25     DO 30 I=1,6
 30     LM(I)=2*IX(I,II)+NT3
        LM(3)=3*IX(3,II)-1
 35     I1=LM(1)
        I2=LM(2)
        I3=LM(3)
        I4=LM(4)
        I5=LM(5)
        I6=LM(6)
        Q(I1)=Q(I1)-WT/12.
        Q(I2)=Q(I2)-WT/12.
        Q(I3)=Q(I3)-WT/12.
        Q(I4)=Q(I4)-WT/4.
        Q(I5)=Q(I5)-WT/4.
        Q(I6)=Q(I6)-WT/4.
 40   CONTINUE
      DO 45 I=1,NDOF
 45   Q(I)=Q(I)/FLOAT(NSTEP)
C
      REWIND 9
C
C                          FORM STIFFNESS MATRIX IN BLOCKS
C
      IMAT=0
      KSHIFT=0
      NUMBLK=1
      NM=ND
      NL=1
      THETA=DELTA*PI/180.
      SLIP=TAN(THETA)
      SLIP=1.1*SLIP
 50   WRITE (6,310) KSHIFT,NUMBLK,NM,NL,ND
      DO 290 KI=1,NELEMNT
        NT=IX(7,KI)
        MT=IX(8,KI)
        ENORML=KN
        ES=KS
        IF (NT.LE.2) GO TO 85
        IF (ISTOP.LE.0) GO TO 85
        IF (MT.NE.IMAT) GO TO 55
        GO TO 85
 55     IMAT=MT
        CALL READMS (4,PROP,319,MT)
        ANISO=PROP(1)
        FRATIO=PROP(3)
        MO=3
        DO 60 I=1,7
          MO=MO+1
 60     XP(I)=PROP(MO)
        DO 65 I=1,11
        DO 65 J=1,7
          MO=MO+1
 65     EP(I,J)=PROP(MO)
        DO 70 I=1,11
        DO 70 J=1,7
          MO=MO+1
```

```
   70   SEP(I,J)=PROP(MO)
        DO 75 I=1,I1
        DO 75 J=1,7
           MO=MO+1
   75   PSNR(I,J)=PROP(MO)
        DO 80 I=1,I1
        DO 80 J=1,7
           MO=MO+1
   80   SPSNR(I,J)=PROP(MO)
        NP=INT(PROP(319))
   85   CONTINUE
C
C                 SEARCH FOR ELEMENTS BELONG TO THIS LAYER
C
        GO TO (90,120,200,200,200), NT
C
C                            TYPE I ELEMENTS
C
   90   DO 95 I=1,2
   95   LM(I)=3*IX(I,KI)-3
        DO 100 I=1,2
           IF (LM(I)+1.LT.NL) GO TO 100
           IF (LM(I)+1.LE.NM) GO TO 105
  100   CONTINUE
        GO TO 290
  105   I1=IX(1,KI)
        I2=IX(2,KI)
        CALL RING (KI,RADIUS,EI,YCEN,X(I1),X(I2),Y(I1),Y(I2),SR)
C
C                          JOIN TYPE I ELEMENTS
C
        DO 115 I=1,2
        DO 115 K=1,3
           II=LM(I)+K-KSHIFT
           IF (II.LE.0.OR.II.GT.MD) GO TO 115
           KK=3*I-3+K
           DO 110 J=1,2
           DO 110 L=1,3
              JJ=LM(J)+L-KSHIFT-II+1
              IF (JJ.LE.0) GO TO 110
              LL=3*J-3+L
              A(JJ,II)=A(JJ,II)+SR(KK,LL)
  110      CONTINUE
  115   CONTINUE
        GO TO 290
C
C                          TYPE II ELEMENTS
C
  120   DO 125 I=1,2
           LM(I)=3*IX(I,KI)-3
  125   CONTINUE
        DO 130 I=3,4
  130   LM(I)=2*IX(I,KI)-2+NT3
        DO 135 I=1,4
           IF (LM(I)+1.LT.NL) GO TO 135
           IF (LM(I)+1.LE.NM) GO TO 140
  135   CONTINUE
        GO TO 290
  140   I1=IX(1,KI)
        I2=IX(2,KI)
```

```
      EN=ENORML
      YC=0.5*(Y(I1)+Y(I2))
      CALL READMS (1,TEMP,17,KI)
      CALL READMS (1,TEMP,17,KI+1)
      IF (INTER.LE.0) GO TO 160
      IF (YC.GT.H2) GO TO 160
      IF (YC.GT.H1.AND.NCYCLE.EQ.1) GO TO 160
      SIGMA=(TEMP(1)+TEMP(1))/2.
      TAU=(TEMP(2)+TEMP(2))/2.
      IF (ABS(SIGMA).LT.GAMA(MT)) GO TO 160
      IF (SIGMA+GAMA(MT)) 155,145,145
  145 RO=TAU/SIGMA
      RATIO=ABS(RO)
      IF (RATIO.GT.SLIP) GO TO 150
      GO TO 160
  150 ES=1000.0
      WRITE (6,320) KI
      GO TO 165
  155 ES=1000.0
      EN=ES
      WRITE (6,315) KI
      GO TO 165
      ES=100.0
      EN=ES
      GO TO 165
  160 EN=ENORML
  165 CALL JOINT (KI,YCEN,EN,ES,X(I1),X(I2),Y(I1),Y(I2),TH1,TH2,SKJT)
      IX(5,KI)=INT(TH1)
      IX(6,KI)=INT(TH2)
      TEMP(3)=EN
      TEMP(4)=ES
      CALL WRITMS (1,TEMP,17,KI)
C
C                          JOIN TYPE II ELEMENT
C
      DO 175 I=1,4
      DO 175 K=1,2
        II=LM(I)+K-KSHIFT
        IF (II.LE.0.OR.II.GT.ND) GO TO 175
        KK=2*I-2+K
        DO 170 J=1,4
        DO 170 L=1,2
          JJ=LM(J)+L-KSHIFT-II+1
          LL=2*J-2+L
          IF (JJ.LE.0) GO TO 170
          A(JJ,II)=A(JJ,II)+SKJT(KK,LL)
  170   CONTINUE
  175 CONTINUE
C
      DO 180 I=1,2
  180 LM(I)=3*IX(I,KI+1)-3
      DO 185 I=3,4
  185 LM(I)=2*IX(I,KI+1)-2+NP3
      I1=IX(1,KI+1)
      I2=IX(2,KI+1)
      CALL JOINT (KI+1,YCEN,EN,ES,X(I1),X(I2),Y(I1),Y(I2),TH1,TH2,SKJ
     1T)
      IX(5,KI+1)=INT(TH1)
      IX(6,KI+1)=INT(TH2)
      TEMP(3)=EN
```

```
      TEMP(4)=ES
      CALL WRITMS (1,TEMP,17,KI+1)
      DO 195 I=1,4
      DO 195 K=1,2
         II=LM(I)+K-KSHIFT
         IF (II.LE.0.OR.II.GT.ND) GO TO 195
         KK=2*I-2+K
         DO 190 J=1,4
         DO 190 L=1,2
            JJ=LM(J)+L-KSHIFT-II+1
            LL=2*J-2+L
            IF (JJ.LE.0) GO TO 190
            A(JJ,II)=A(JJ,II)+SKJT(KK,LL)
  190    CONTINUE
  195 CONTINUE
      KI=KI+1
C
      GO TO 290
C
C                         TRIANGULAR ELEMENT TYPE III
C
  200 MK=NT-2
      GO TO (205,215,220), MK
  205 DO 210 I=1,6
  210 LM(I)=2*IX(I,KI)-2+NT3
      GO TO 230
  215 LM(1)=3*IX(1,KI)-3
      LM(2)=3*IX(2,KI)-3
      LM(3)=2*IX(3,KI)-2+NT3
      LM(4)=3*IX(4,KI)-3
      LM(5)=2*IX(5,KI)-2+NT3
      LM(6)=2*IX(6,KI)-2+NT3
      GO TO 230
  220 DO 225 I=1,6
  225 LM(I)=2*IX(I,KI)-2+NT3
      LM(3)=3*IX(3,KI)-3
  230 DO 235 I=1,6
         IF (LM(I)+1.LT.NL) GO TO 235
         IF (LM(I)+1.LE.NM) GO TO 240
  235 CONTINUE
      GO TO 290
  240 I1=IX(1,KI)
      I2=IX(2,KI)
      I3=IX(3,KI)
      YC=(Y(I1)+Y(I2)+Y(I3))/3.
C
C                               TYPE III ELEMENT
C
      IF (ISTOP.LE.0) GO TO 270
      IF (YC.GT.H2) GO TO 265
      IF (YC.GT.H1.AND.NCYCLE.EQ.1) GO TO 270
      CALL READMS (1,TEMP,17,KI)
      STRESSX=ABS(TEMP(7))
      TAUOCT=ABS(TEMP(8))
      TAUF=STRESSX*FRATIO
      RATIO=TAUOCT/TAUF
      IF (RATIO.GT.1.0) WRITE (6,325) RATIO,KI
      DO 245 LI=1,10
         R1=FLOAT(LI-1)/10.
         R2=FLOAT(LI)/10.
```

```
         LJ=11-LI+1
         IF (RATIO.GT.R1.AND.RATIO.LE.R2) GO TO 250
  245 CONTINUE
      LJ=2
C
C   * LJ * IS THE ET VS. SIGMA3 CURVE NO. SELECTED BASED ON STRESS RAT
C                  NOW SELECT INTERVAL OF COPFINING?PRESSURE
C
  250 IF (STRESSX.LT.XP(1).OR.STRESSX.GT.XP(NP)) WRITE (6,330) STRESS
     1 X,KI
      DO 255 LK=2,NP
         LI=LK
         IF (STRESSX.GE.XP(LK-1).AND.STRESSX.LE.XP(LK)) GO TO 260
  255 CONTINUE
      LI=NP
  260 YDP1=SEP(LJ,LI-1)
      YDPN1=SPSNR(LJ,LI-1)
      YDP2=SEP(LJ,LI)
      YDPN2=SPSNR(LJ,LI)
      X1=XP(LI-1)
      X2=XP(LI)
      Y1=EP(LJ,LI-1)
      YN1=PSNR(LJ,LI-1)
      Y2=EP(LJ,LI)
      YN2=PSNR(LJ,LI)
      PP=STRESSX
      ET1=ORDINET(YDP1,YDP2,X1,X2,Y1,Y2,PP)
      PSN1=ORDINET(YDPN1,YDPN2,X1,X2,YN1,YN2,PP)
      YDP1=SEP(LJ-1,LI-1)
      YDPN1=SPSNR(LJ-1,LI-1)
      YDP2=SEP(LJ-1,LI)
      YDPN2=SPSNR(LJ-1,LI)
      Y1=EP(LJ-1,LI-1)
      YN1=PSNR(LJ-1,LI-1)
      Y2=EP(LJ-1,LI)
      YN2=PSNR(LJ-1,LI)
      ET2=ORDINET(YDP1,YDP2,X1,X2,Y1,Y2,PP)
      PSN2=ORDINET(YDPN1,YDPN2,X1,X2,YN1,YN2,PP)
      EY=ET2+(ET1-ET2)*(RATIO-R1)
      NUEY=PSN2+(PSN1-PSN2)*(RATIO-R1)
      EX=ANISO*EY
      NUEX=NUEY*EX/EY
      GO TO 275
  265 EY=E(MT)/1000.0
      EX=ANISO*EY
      NUEY=NUE(MT)
      NUEX=NUEY*EX/EY
      GO TO 275
  270 EY=E(MT)
      EX=ANISO*EY
      NUEY=NUE(MT)
      NUEX=NUEY*EX/EY
  275 CALL TRANGLE (KI,NT,EX,EY,NUEX,NUEY,SK)
      DO 285 I=1,6
      DO 285 K=1,2
         II=LM(I)+K-KSHIFT
         IF (II.LE.0.OR.II.GT.ND) GO TO 285
         KK=I+(K-1)*6
         DO 280 J=1,6
         DO 280 L=1,2
```

```
                JJ=LM(J)+L-KSHIFT+1-II
                IF (JJ.LE.0) GO TO 280
                LL=J+(L-1)*6
                A(JJ,II)=A(JJ,II)+SK(KK,LL)
  280         CONTINUE
  285      CONTINUE
  290 CONTINUE
C
C                 WRITE STIFFNESS EQUATUIONS ON TAPE 10
C
C                 WRITE(10) ((A(M,N),M=1,MBAND),N=1,ND)
C
      CALL ZERO (ARRAY,NSIZE)
      NX=NL-1
      DO 300 M=1,NSIZE
         DO 295 N=1,NSIZE
  295    ARRAY(N)=A(M,N)
         CALL WRITMS (10,ARRAY,NSIZE,NX+M)
  300 CONTINUE
      IF (NM.GE.NDOF) GO TO 305
      CALL NULLMAT (A,NSIZE,NSIZE)
      KSHIFT=KSHIFT+ND
      IF (NUMBLK.EQ.1) ND=ND-NDINC
      NUMBLK=NUMBLK+1
      NM=NM+ND
      NL=NM-ND+1
      GO TO 50
  305 IF (NT3.GT.0) ND=ND+NDINC
      RETURN
C
  310 FORMAT (5X, 9HKSHIFT = ,I8,5X, 9HNUMBLK = ,I5, 8H   NM = ,I5,
     18H   NL = ,I5, 7H  ND = ,I5)
  315 FORMAT (5X, 24HINTERACTION ELEMENT NO. ,I5,3X, 21HHAS FAILED IN TE
     1NSION)
  320 FORMAT (5X, 26HINTERACTION ELEMENT NO. = ,I5,3X, 31H HAS FAILED IN
     1 EXCESSIVE  SHEAR)
  325 FORMAT (10X, 41HSTRESS RATIO IS GREATER THAN 1 , RATIO = ,F6.2,5X,
     1 17HIN ELEMENT NO. = ,I5)
  330 FORMAT (10X, 56HCONFINING PRESSURE IS OUT OF RANGE OF SIGMA3, STRE
     1SSX = ,E12.2,5X, 17HIN ELEMENT NO. = ,I5)
C
      END
      SUBROUTINE TRANGLE (K,NT,EX,EY,NUEX,NUEY,SK)
      COMMON NNODES,NELEMNT,NDOF,MBAND,ND,NT3,ISTOP,NCYCLE,LAYERS,ISTEP,
     1NSTEP,NT12,ETA,NT1,NT2,NDIMENSN,IFLAG,NSIZE,NCODE(550),X(550),Y(550
     2),JNDX(51),ANLSIS,IX(8,250),AREAA(250),INDX(250),INDEX(250),GAMA(2
     35),ZAI(3)
      COMMON /1/ E,NUE,RADIUS,XCEN,YCEN,EI,KN,KS,H1,H2,INTER
      COMMON /2/ D
      DATA PLSTRS,PLSTRN/6HPLSTRS,6HPLSTRN/
      REAL NUEX,NUEY,NUENS
      DIMENSION SK(12,12), E(10), NUE(10), D(10,10)
      DIMENSION DD(3,3), DMAT(90)
      REAL KN,KS
C
C     ***********************************************************************
C             GENERATE STIFFNESS MATRIX FOR TRIANGULAR ELEMENTS
C
      AA=AREAA(K)
      MT=IX(8,K)
```

```
      TA=1./AA
C
C                              TYPE III ELEMENL
C
      L=IX(1,K)
      M=IX(2,K)
      N=IX(3,K)
      X21=X(M)-X(L)
      X13=X(L)-X(N)
      X32=X(N)-X(M)
      Y12=Y(L)-Y(M)
      Y23=Y(M)-Y(N)
      Y31=Y(N)-Y(L)
      IF (ISTOP.LE.0) GO TO 5
      GO TO 10
    5 D11=D(MT,1)
      D12=D(MT,2)
      D13=D(MT,3)
      D21=D(MT,4)
      D22=D(MT,5)
      D23=D(MT,6)
      D31=D(MT,7)
      D32=D(MT,8)
      D33=D(MT,9)
      GO TO 15
C
   10 IF (ANLSIS.EQ.PLSTRS) CALL PLNSTRS (EX,EY,NUEX,NUEY,DD)
      IF (ANLSIS.EQ.PLSTRN) CALL VALUES (EX,EY,NUEX,NUEY,DD)
      D11=DD(1,1)
      D12=DD(1,2)
      D13=DD(1,3)
      D21=DD(2,1)
      D22=DD(2,2)
      D23=DD(2,3)
      D31=DD(3,1)
      D32=DD(3,2)
      D33=DD(3,3)
   15 A1=D11*Y23+D13*X32
      A2=D31*Y23+D33*X32
      A3=D11*Y31+D13*X13
      A4=D31*Y31+D33*X13
      A5=D11*Y12+D13*X21
      A6=D31*Y12+D33*X21
      A7=D12*X32+D13*Y23
      A8=D32*X32+D33*Y23
      A9=D12*X13+D13*Y31
      A10=D32*X13+D33*Y31
      A11=D12*X21+D13*Y12
      A12=D32*X21+D33*Y12
C
      B1=D21*Y23+D23*X32
      B2=D21*Y31+D23*X13
      B3=D21*Y12+D23*X21
      B4=D22*X32+D23*Y23
      B5=D22*X13+D23*Y31
      B6=D22*X21+D23*Y12
C
      C11=Y23*A1+X32*A2
      C12=Y23*A3+X32*A4
      C13=Y23*A5+X32*A6
```

```
      C14=Y23*A7+X32*A8
      C15=Y23*A9+X32*A10
      C16=Y23*A11+X32*A12
C
      C21=Y31*A1+X13*A2
      C22=Y31*A3+X13*A4
      C23=Y31*A5+X13*A6
      C24=Y31*A7+X13*A8
      C25=Y31*A9+X13*A10
      C26=Y31*A11+X13*A12
C
      C31=Y12*A1+X21*A2
      C32=Y12*A3+X21*A4
      C33=Y12*A5+X21*A6
      C34=Y12*A7+X21*A8
      C35=Y12*A9+X21*A10
      C36=Y12*A11+X21*A12
C
      C41=X32*B1+Y23*A2
      C42=X32*B2+Y23*A4
      C43=X32*B3+Y23*A6
      C44=X32*B4+Y23*A8
      C45=X32*B5+Y23*A10
      C46=X32*B6+Y23*A12
C
      C51=X13*B1+Y31*A2
      C52=X13*B2+Y31*A4
      C53=X13*B3+Y31*A6
      C54=X13*B4+Y31*A8
      C55=X13*B5+Y31*A10
      C56=X13*B6+Y31*A12
C
      C63=X21*B3+Y12*A6
      C61=X21*B1+Y12*A2
      C62=X21*B2+Y12*A4
      C64=X21*B4+Y12*A8
      C65=X21*B5+Y12*A10
      C66=X21*B6+Y12*A12
C
      IF (K.GT.NT12.AND.K.LE.(NT12+NT1/2)) GO TO 20
      GO TO 25
   20 ETTA=ETA/2.
      ETATWO=1.-2.*ETTA
      ETAFOUR=1.-4.*ETTA
      ETA2=ETATWO*ETATWO
      ETA4=ETAFOUR*ETAFOUR
      GO TO 30
   25 ETTA=0.0
      ETAFOUR=1.0
      ETATWO=ETAFOUR
      ETA4=1.0
      ETA2=ETA4
C
C
   30 SK(1,1)=C11*TA*(1.+4.*ETTA/3.+4.*ETTA*ETTA)/(ETATWO*ETATWO*4.)
      SK(2,1)=-C12*TA/(12.*ETAFOUR)
      SK(1,2)=SK(2,1)
C
      SK(1,3)=C13*TA*(-0.333+2.*ETTA+8.*ETTA*ETTA)/(4.*ETATWO)
      SK(3,1)=SK(1,3)
```

```
      SK(2,3)=-C23*TA/12.
      SK(3,2)=SK(2,3)
      SK(2,2)=C22*TA/(4.*ETA4)
      SK(3,3)=C33*TA*(1.+8.*ETTA/3.+16.*ETTA*ETTA)/4.
C
      SK(1,4)=TA*(C11*2.*ETTA+C12*(4./3.+8.*ETTA/3.))/(4.*ETA2*ETATWO)
      SK(4,1)=SK(1,4)
      SK(2,4)=C21*TA/(3.*ETA2*ETAFOUR)
      SK(4,2)=SK(2,4)
      SK(3,4)=4.*(C31+C32)*TA*ETTA/(3.*ETA2)
      SK(4,3)=SK(3,4)
      SK(4,4)=(2.*C11+C12+C21+2.*C22)*TA/(3.*ETA2*ETA2)
C
      SK(1,5)=2.*ETTA*(C11+C13)*TA/(3.*ETATWO*ETAFOUR)
      SK(5,1)=SK(1,5)
      SK(2,5)=C23*TA/(3.*ETA4)
      SK(5,2)=SK(2,5)
      SK(3,5)=(C32*(1.+4.*ETTA)+C33*4.*ETTA)*TA/(3.*ETAFOUR)
      SK(5,3)=SK(3,5)
      SK(4,5)=TA*(C12+2.*C13+C22+C23)/(3.*ETAFOUR*ETA2)
      SK(5,4)=SK(4,5)
      SK(5,5)=TA*(2.*C22+C23+C32+2.*C33)/(3.*ETA4)
C
      SK(1,6)=TA*(C13+2.*ETTA*(C11+C13))/(3.*ETATWO*ETAFOUR)
      SK(6,1)=SK(1,6)
      SK(2,6)=0.0
      SK(6,2)=SK(2,6)
      SK(3,6)=TA*(C31+4.*ETTA*(C31+C33))/(3.*ETAFOUR)
      SK(6,3)=SK(3,6)
      SK(4,6)=TA*(C11+C13+C21+2.*C23)/(3.*ETAFOUR*ETA2)
      SK(6,4)=SK(4,6)
      SK(5,6)=TA*(2.*C21+C23+C31+C33)/(3.*ETA4)
      SK(6,5)=SK(5,6)
      SK(6,6)=TA*(2.*C11+C13+C31+2.*C33)/(3.*ETA4)
C
      SK(1,7)=C14*TA*(1.+4.*ETTA/3.+4.*ETTA*ETTA)/(4.*ETA2)
      SK(7,1)=SK(1,7)
      SK(2,7)=-C24*TA/(12.*ETAFOUR)
      SK(7,2)=SK(2,7)
      SK(3,7)=-C34*TA*(1.-6.*ETTA-24.*ETTA*ETTA)/(12.*ETATWO)
      SK(7,3)=SK(3,7)
      SK(4,7)=TA*(2.*C14*ETTA+C24*(1.+2.*ETTA))/(3.*ETA2*ETATWO)
      SK(7,4)=SK(4,7)
      SK(5,7)=2.*ETTA*TA*(C24+C34)/(3.*ETATWO*(1.+2.*ETTA))
      SK(7,5)=SK(5,7)
      SK(6,7)=TA*(2.*C14*ETTA+C34*(1.+2.*ETTA))/(3.*ETATWO*ETAFOUR)
      SK(7,6)=SK(6,7)
      SK(7,7)=TA*C44*(1.+4.*ETTA/3.+4.*ETTA*ETTA)/(4.*ETA2)
C
      SK(1,8)=TA*C15*(-1.+2.*ETTA)/(12.*ETAFOUR*ETATWO)
      SK(8,1)=SK(1,8)
      SK(2,8)=TA*C25/(4.*ETA4)
      SK(8,2)=SK(2,8)
      SK(3,8)=-C35*TA/12.
      SK(8,3)=SK(3,8)
      SK(4,8)=C15*TA/(3.*ETAFOUR*ETA2)
      SK(8,4)=SK(4,8)
      SK(5,8)=C35*TA/(3.*ETA4)
      SK(8,5)=SK(5,8)
      SK(6,8)=0.0
```

```
      SK(8,6)=SK(6,8)
      SK(7,8)=-C45*TA/(12.*ETAFOUR)
      SK(8,7)=SK(7,8)
      SK(8,8)=C55*TA/(4.*ETA4)
C
      SK(1,9)=-C16*TA*(1.-2.*ETTA-8.*ETTA*ETTA)/(12.*ETATWO)
      SK(9,1)=SK(1,9)
      SK(2,9)=-C26*TA/12.
      SK(9,2)=SK(2,9)
      SK(3,9)=C36*TA*(1.+8.*ETTA/3.+16.*ETTA*ETTA)/4.
      SK(9,3)=SK(3,9)
      SK(4,9)=4.*ETTA*TA*(C16+C26)/(3.*ETA2)
      SK(9,4)=SK(4,9)
      SK(5,9)=TA*(C26*(1.+4.*ETTA)+C36*4.*ETTA)/(3.*ETAFOUR)
      SK(9,5)=SK(5,9)
      SK(6,9)=TA*(C16*(1.+4.*ETTA)+C36*4.*ETTA)/(3.*ETAFOUR)
      SK(9,6)=SK(6,9)
      SK(7,9)=-TA*C46*(1.-6.*ETTA-24.*ETTA*ETTA)/(12.*ETATWO)
      SK(9,7)=SK(7,9)
      SK(8,9)=-C56*TA/12.
      SK(9,8)=SK(8,9)
      SK(9,9)=C66*TA*(1.+8.*ETTA/3.+16.*ETTA*ETTA)/4.
C
      SK(1,10)=TA*(C14*2.*ETTA+C15*(1.+2.*ETTA))/(3.*ETATWO*ETA2)
      SK(10,1)=SK(1,10)
      SK(2,10)=C24*TA/(3.*ETAFOUR*ETA2)
      SK(10,2)=SK(2,10)
      SK(3,10)=4.*TA*ETTA*(C34+C35)/(3.*ETA2)
      SK(10,3)=SK(3,10)
      SK(4,10)=TA*(2.*C14+C15+C24+2.*C25)/(3.*ETA2*ETA2)
      SK(10,4)=SK(4,10)
      SK(5,10)=TA*(C24+C25+2.*C34+C35)/(3.*ETAFOUR*ETA2)
      SK(10,5)=SK(5,10)
      SK(6,10)=TA*(C14+C15+C34+2.*C35)/(3.*ETA2*ETAFOUR)
      SK(10,6)=SK(6,10)
      SK(7,10)=TA*(2.*ETTA*C44+C45*(1.+2.*ETTA))/(3.*ETA2*ETATWO)
      SK(10,7)=SK(7,10)
      SK(8,10)=C54*TA/(3.*ETAFOUR*ETA2)
      SK(10,8)=SK(8,10)
      SK(9,10)=4.*ETTA*TA*(C64+C65)/(3.*ETA2)
      SK(10,9)=SK(9,10)
      SK(10,10)=TA*(2.*C44+C45+C54+2.*C55)/(3.*ETA4*ETA4)
C
      SK(1,11)=2.*ETTA*TA*(C15+C16)/(3.*ETATWO*ETAFOUR)
      SK(11,1)=SK(1,11)
      SK(2,11)=C26*TA/(3.*ETA4)
      SK(11,2)=SK(2,11)
      SK(3,11)=TA*(C35*(1.+4.*ETTA)+4.*ETTA*C36)/(3.*ETAFOUR)
      SK(11,3)=SK(3,11)
      SK(4,11)=TA*(C15+2.*C16+C25+C26)/(3.*ETAFOUR*ETA2)
      SK(11,4)=SK(4,11)
      SK(5,11)=TA*(2.*C25+C26+C35+2.*C36)/(3.*ETA4)
      SK(11,5)=SK(5,11)
      SK(6,11)=TA*(2.*C15+C16+C35+C36)/(3.*ETA4)
      SK(11,6)=SK(6,11)
      SK(7,11)=TA*2.*ETTA*(C45+C46)/(3.*ETATWO*ETAFOUR)
      SK(11,7)=SK(7,11)
      SK(8,11)=TA*C56/(3.*ETA4)
      SK(11,8)=SK(8,11)
      SK(9,11)=TA*(C65*(1.+4.*ETTA)+4.*ETTA*C66)/(3.*ETAFOUR)
```

```
      SK(11,9)=SK(9,11)
      SK(10,11)=TA*(C45+2.*C46+C55+C56)/(3.*ETAFOUR*ETA2)
      SK(11,10)=SK(10,11)
      SK(11,11)=TA*(2.*C55+C56+C65+2.*C66)/(3.*ETA4)
C
      SK(1,12)=TA*(C14*2.*ETTA+C16*(1.+2.*ETTA))/(3.*ETATWO)
      SK(12,1)=SK(1,12)
      SK(2,12)=0.0
      SK(12,2)=SK(2,12)
      SK(3,12)=TA*(C34*(1.+4.*ETTA)+C36*4.*ETTA)/(3.*ETAFOUR)
      SK(12,3)=SK(3,12)
      SK(4,12)=TA*(C14+C16+C24+2.*C26)/(3.*ETAFOUR*ETA2)
      SK(12,4)=SK(4,12)
      SK(5,12)=TA*(2.*C24+C26+C34+C36)/(3.*ETA4)
      SK(12,5)=SK(5,12)
      SK(6,12)=TA*(2.*C14+C16+C34+2.*C36)/(3.*ETA4)
      SK(12,6)=SK(6,12)
      SK(7,12)=TA*(2.*ETTA*C44+(1.+2.*ETTA)*C46)/(3.*ETATWO*ETAFOUR)
      SK(12,7)=SK(7,12)
      SK(8,12)=0.0
      SK(12,8)=SK(8,12)
      SK(9,12)=TA*(C64*(1.+4.*ETTA)+C66*4.*ETTA)/(3.*ETAFOUR)
      SK(12,9)=SK(9,12)
      SK(10,12)=TA*(C44+C46+C54+2.*C56)/(3.*ETAFOUR*ETA2)
      SK(12,10)=SK(10,12)
      SK(11,12)=TA*(2.*C54+C56+C64+C66)/(3.*ETA4)
      SK(12,11)=SK(11,12)
      SK(12,12)=TA*(2.*C44+C46+C64+2.*C66)/(3.*ETA4)
C
C
      MO=0
      DO 45 I=1,3
      DO 45 J=1,3
         MO=MO+1
         IF (ISTOP) 35,35,40
   35    DMAT(MO)=D(MT,MO)
         GO TO 45
   40    DMAT(MO)=DD(I,J)
   45 CONTINUE
      DMAT(10)=NUEX
      DMAT(11)=NUEY
      MO=11
      DO 55 I=1,12
         DO 50 J=I,12
            MO=MO+1
   50    DMAT(MO)=SK(I,J)
   55 CONTINUE
      CALL WRITMS (3,DMAT,89,K)
      RETURN
C
      END
      SUBROUTINE MATMULT (A,B,C,M,N)
      DIMENSION A(M,N), B(M,N), C(M,N)
C
C           MATRIX * A * IS MULTIPLIED WITH * B * AND STORED IN * C *
C
      DO 10 I=1,M
      DO 10 J=1,N
         SUM=0.0
         DO 5 K=1,M
```

```
  5    SUM=SUM+A(I,K)*B(K,J)
 10 C(I,J)=SUM
    RETURN
C
    ENTRY MATRMUL
C
C    TRANSPOSE OF * A * IS MULTIPLIED WITH * B * AND STORED INTO * C *
C
    DO 20 I=1,M
    DO 20 J=1,M
       SUM=0.0
       DO 15 K=1,M
 15    SUM=SUM+A(K,I)*B(K,J)
 20 C(I,J)=SUM
    RETURN
C
    END
    SUBROUTINE VALUES (EX,EY,NUEX,NUEY,DD)
C
C           CALCULATE = DD = MATRIX FOR PLANE STRAIN SITUATION
C
    DIMENSION DD(3,3)
    REAL NUEX,NUEY
    RN=EX/EY
    RM=0.5/(1.+NUEY)
    C=EY/((1.+NUEX)*(1.-NUEX-2.*RN*NUEY*NUEY))
    DD(1,1)=C*RN*(1.-RN*NUEY*NUEY)
    DD(2,1)=C*RN*(1.+NUEX)*NUEY
    DD(1,2)=DD(2,1)
    DD(2,2)=C*(1.-NUEX*NUEX)
    DD(3,3)=C*RM*(1.+NUEX)*(1.-NUEX-2.*RN*NUEY*NUEY)
    DD(3,2)=0.0
    DD(3,1)=DD(3,2)
    DD(2,3)=DD(3,1)
    DD(1,3)=DD(2,3)
    RETURN
C
    END
    SUBROUTINE PLNSTRS (EX,EY,NUEX,NUEY,DD)
C
C           CALCULATE = DD = MATRIX FOR PLANE STRESS SITUATION
C
    DIMENSION DD(3,3)
    REAL NUEX,NUEY
    RN=EX/EY
    RM=0.5/(1.+NUEY)
    C=EY/(1.-RN*NUEY*NUEY)
    DD(1,1)=C*RN
    DD(2,1)=C*RN*NUEY
    DD(1,2)=DD(2,1)
    DD(2,2)=C
    DD(3,3)=C*RM*(1.-RN*NUEY*NUEY)
    DD(3,2)=0.0
    DD(3,1)=DD(3,2)
    DD(2,3)=DD(3,1)
    DD(1,3)=DD(2,3)
    RETURN
C
    END
    FUNCTION ORDINET(A,B,C,D,E,F,P)
```

```
C
C          DETERMINE TANGENT MODULUS USING SPLINE FUNCTION
C
      HJ=D-C
      B1=D-P
      B2=P-C
      A1=B1*B1*B1
      A2=B2*B2*B2
      T1=A1*A/(6.*HJ)
      T2=A2*B/(6.*HJ)
      T3=(E-A*HJ*HJ/6.)*(D-P)/HJ
      T4=(F-B*HJ*HJ/6.)*(P-C)/HJ
      ORDINET=T1+T2+T3+T4
      RETURN
C
      END
      SUBROUTINE RING (KI,R,EI,YCEN,X1,X2,Y1,Y2,SK)
      DIMENSION SR(6,6), SK(6,6)
      DIMENSION TL(6,6), TC(6,6)
      DIMENSION TMP(36)
      REAL NUE
C
C                STIFFNESS MATRIX FUR RING ELEMENTS
C
      SPAN=SQRT((X1-X2)**2+(Y1-Y2)**2)
      BETA=SPAN/R
      BETA2=2.*BETA
      SN=SIN(BETA)
      CS=COS(BETA)
      SN2=SIN(BETA2)
      A=BETA-SN
      B=CS+SN*SN/2.-1.
      C=1.5*BETA-2.*SN+SN2/4.
      D=0.5*BETA-SN2/4.
      E=CS-1.
      AA=E*E/BETA-D
      BB=B-A*E/BETA
      CC=A*D-B*E
      DD=A*A/BETA-C
      EE=C*E-A*B
      FF=B*B-C*D
      G=B*(B-2.*A*E/BETA)+C*(E*E/BETA-D)+A*A*D/BETA
      FT=EI/((R**3)*G)
      SR(1,1)=FT*AA
      SR(1,2)=FT*BB
      SR(2,1)=SR(1,2)
      SR(1,3)=FT*CC*R/BETA
      SR(3,1)=SR(1,3)
      SR(2,2)=FT*DD
      SR(2,3)=FT*EE*R/BETA
      SR(3,2)=SR(2,3)
      SR(3,3)=FT*FF*R*R/BETA
      SR(4,1)=-FT*(AA*CS+BB*SN)
      SR(1,4)=SR(4,1)
      SR(4,2)=-FT*(BB*CS+DD*SN)
      SR(2,4)=SR(4,2)
      SR(4,3)=-FT*(CC*CS+EE*SN)*R/BETA
      SR(3,4)=SR(4,3)
      SR(1,5)=FT*(AA*SN-BB*CS)
      SR(5,1)=SR(1,5)
```

```
      SR(5,2)=FT*(BB*SN-DD*CS)
      SR(2,5)=SR(5,2)
      SR(5,3)=FT*(CC*SN-EE*CS)*R/BETA
      SR(3,5)=SR(5,3)
      SR(6,1)=FT*(AA*(CS-1.)+BB*SN-CC/BETA)*R
      SR(1,6)=SR(6,1)
      SR(6,2)=FT*(BB*(CS-1.)+DD*SN-EE/BETA)*R
      SR(2,6)=SR(6,2)
      SR(6,3)=FT*(CC*(CS-1.)+EE*SN-FF)*R*R/BETA
      SR(3,6)=SR(6,3)
      SR(4,4)=SR(1,1)
      SR(4,5)=-SR(1,2)
      SR(5,4)=SR(4,5)
      SR(4,6)=SR(1,3)
      SR(6,4)=SR(4,6)
      SR(5,5)=SR(2,2)
      SR(5,6)=-SR(2,3)
      SR(6,5)=SR(5,6)
      SR(6,6)=SR(3,3)
C
C                   TRANSFORMATION OF COORDINATE SYSTEM
C
      CALL NULLMAT (TL,6,6)
      CALL NULLMAT (TC,6,6)
      T1=Y1-YCEN
      T2=Y2-YCEN
      IF (X1.EQ.0.0) X1=0.00001
      IF (X2.EQ.0.0) X2=0.00001
      TN1=T1/X1
      TN2=T2/X2
      PHI1=ATAN(TN1)
      PHI2=ATAN(TN2)
      CS1=COS(PHI1)
      SN1=SIN(PHI1)
      CS2=COS(PHI2)
      SN2=SIN(PHI2)
      TL(1,1)=-SN1
      TL(1,2)=+CS1
      TL(2,1)=-CS1
      TL(2,2)=-SN1
      TL(4,4)=-SN2
      TL(4,5)=+CS2
      TL(5,4)=-CS2
      TL(5,5)=-SN2
      TL(6,6)=1.0
      TL(3,3)=TL(6,6)
      CALL MATMULT (SR,TL,TC,6,6)
      CALL MATRMUL (TL,TC,SK,6,6)
      MO=0
      DO 5 I=1,6
      DO 5 J=1,6
         MO=MO+1
         TMP(MO)=SK(I,J)
    5 CONTINUE
      CALL WRITMS (7,TMP,MO,KI)
      RETURN
C
      END
      SUBROUTINE JOINT (KI,YCEN,EN,ES,X1,X2,Y1,Y2,TH1,TH2,SKJT)
      DIMENSION SKJT(8,8), TL(8,8), TC(8,8), TMP(64)
```

```
      EQUIVALENCE (TC(1,1),TMP(1))
C
C                     STIFFNESS MATRIX FOR INTERACTION ELEMEN
C
      PI=22./7.
      SPAN=SQRT((X1-X2)**2+(Y1-Y2)**2)
      T1=Y1-YCEN
      T2=Y2-YCEN
      T1=-T1
      T2=-T2
      IF (T1.EQ.0.0) T1=0.00001
      IF (T2.EQ.0.0) T1=0.00001
      TN1=ABS(X1/T1)
      TN2=ABS(X2/T2)
      THETA1=ATAN(TN1)
      THETA2=ATAN(TN2)
      IF (T1.LT.0.0) THETA1=22./7.-THETA1
      IF (T2.LT.0.0) THETA2=22./7.-THETA2
      TH1=10.*(THETA1*180./PI)
      TH2=10.*(THETA2*180./PI)
      CALL NULLMAT (SKJT,8,8)
      CS=SPAN*ES/6.
      CN=SPAN*EN/6.
      SKJT(1,1)=2.*CS
      SKJT(1,3)=CS
      SKJT(1,5)=-CS
      SKJT(1,7)=-2.*CS
      SKJT(2,2)=2.*CN
      SKJT(2,4)=CN
      SKJT(2,6)=-CN
      SKJT(2,8)=-2.*CN
      SKJT(3,1)=CS
      SKJT(3,3)=2.*CS
      SKJT(3,5)=-2.*CS
      SKJT(3,7)=-CS
      SKJT(4,2)=CN
      SKJT(4,4)=2.*CN
      SKJT(4,6)=-2.*CN
      SKJT(4,8)=-CN
      SKJT(5,1)=-CS
      SKJT(5,3)=-2.*CS
      SKJT(5,5)=2.*CS
      SKJT(5,7)=CS
      SKJT(6,2)=-CN
      SKJT(6,4)=-2.*CN
      SKJT(6,6)=2.*CN
      SKJT(6,8)=CN
      SKJT(7,1)=-2.*CS
      SKJT(7,3)=-CS
      SKJT(7,5)=CS
      SKJT(7,7)=2.*CS
      SKJT(8,2)=-2.*CN
      SKJT(8,4)=-CN
      SKJT(8,6)=CN
      SKJT(8,8)=2.*CN
      CALL NULLMAT (TL,8,8)
      CALL NULLMAT (TC,8,8)
      CS1=COS(THETA1)
      SN1=SIN(THETA1)
      CS2=COS(THETA2)
```

```
      SN2=SIN(THETA2)
      TL(7,7)=CS1
      TL(1,1)=TL(7,7)
      TL(7,8)=SN1
      TL(1,2)=TL(7,8)
      TL(8,7)=-SN1
      TL(2,1)=TL(8,7)
      TL(8,8)=CS1
      TL(2,2)=TL(8,8)
      TL(5,5)=CS2
      TL(3,3)=TL(5,5)
      TL(5,6)=SN2
      TL(3,4)=TL(5,6)
      TL(6,5)=-SN2
      TL(4,3)=TL(6,5)
      TL(6,6)=CS2
      TL(4,4)=TL(6,6)
      CALL MATMULT (SKJT,TL,TC,8,8)
      CALL MATRMUL (TL,TC,SKJT,8,8)
      MO=0
      DO 5 I=1,8
      DO 5 J=1,8
         MO=MO+1
         TMP(MO)=SKJT(I,J)
    5 CONTINUE
      CALL WRITMS (7,TMP,MO,KI)
      RETURN
C
      END
      OVERLAY(ROY,4,0)
C                                    OVERLAY(ROY,4,0)
      PROGRAM STRESS
      COMMON NNODES,NELEMNT,NDOF,MBAND,ND,NT3,ISTOP,NCYCLE,LAYERS,ISTEP,
     1NSTEP,NT12,ETA,NT1,NT2,NDTENSN,IFLAG,NSIZE,NCODE(550),X(550),Y(550
     2),JNDX(51),ANLSIS,IX(8,250),ARENA(250),INDX(250),INDEX(250),GAMA(2
     35),ZAI(3)
      COMMON /1/ E,NUE,RADIUS,XCEN,YCEN,EI,KN,KS,H1,H2,INTER
      COMMON /2/ D
      COMMON /3/ R
      COMMON /4/ MUDU(26),DELTA
      COMMON /5/ Q(1100),LIST(1101)
      REAL NUE1,NUE2,NUE,NUEK,KS,KN
      DIMENSION TEMP(17), R(1100), E(10), NUE(10), D(10,10)
      DIMENSION DMAT(90), U(12), SKJT(12,12), PP(12), TEMP(17)
      DIMENSION IA(50,4), FR(2)
C
C     ******************************************************************
C       THIS ROUTINE COMPUTES STRESS, STRAIN, OCTAHEADRAL STRESSES AND
C      STRAINS FROM COMPUTED NODAL DISPLACEMENTS STORED AS R(I) ON TAPE
C
C     ******************************************************************
C
      PI=22./7.
      CALL ZERO (Q,NDOF)
      IFLAG=0
      THETA=DELTA*PI/180.
      SLIP=TAN(THETA)
      SLIP=1.1*SLIP
C
      DO 100 KEL=1,NELEMNT
```

```
      ITYP=IX(7,KEL)
      MT=IX(8,KEL)
      GO TO (100,5,35,35,35), ITYP
C
C                    TYPE II INTERACTION ELEMENT
C
    5 CALL ZERO (U,12)
      DO 10 I=1,2
      DO 10 J=1,2
         K=(I-1)*2+J
         N=3*IX(I,KEL)-3+J
         IA(KEL,I)=N
         U(K)=R(N)
   10 CONTINUE
      DO 15 I=3,4
      DO 15 J=1,2
         K=(I-1)*2+J
         N=2*IX(I,KEL)-2+NT3+J
         IA(KEL,I)=N
         U(K)=R(N)
   15 CONTINUE
      I=IX(1,KEL)
      J=IX(2,KEL)
      SPAN=SQRT((X(I)-X(J))**2+(Y(I)-Y(J))**2)
      TH1=FLOAT(IX(5,KEL))
      TH2=FLOAT(IX(6,KEL))
      THETA1=TH1*PI/1800.
      THETA2=TH2*PI/1800.
      CS1=COS(THETA1)
      SN1=SIN(THETA1)
      CS2=COS(THETA2)
      SN2=SIN(THETA2)
      CALL ZERO (PP,12)
      CALL NULLMAT (SKJT,12,12)
      SKJT(7,7)=CS1
      SKJT(1,1)=SKJT(7,7)
      SKJT(7,8)=SN1
      SKJT(1,2)=SKJT(7,8)
      SKJT(8,7)=-SN1
      SKJT(2,1)=SKJT(8,7)
      SKJT(8,8)=CS1
      SKJT(2,2)=SKJT(8,8)
      SKJT(5,5)=CS2
      SKJT(3,3)=SKJT(5,5)
      SKJT(5,6)=SN2
      SKJT(3,4)=SKJT(5,6)
      SKJT(6,5)=-SN2
      SKJT(4,3)=SKJT(6,5)
      SKJT(6,6)=CS2
      SKJT(4,4)=SKJT(6,6)
      DO 25 I=1,8
         SUM=0.0
         DO 20 K=1,8
   20    SUM=SUM+SKJT(I,K)*U(K)
         PP(I)=SUM
   25 CONTINUE
      T1=-PP(1)-PP(3)+PP(5)+PP(7)
      T2=-PP(2)-PP(4)+PP(6)+PP(8)
      CALL READMS (1,TEMP,17,KEL)
      FIXN=TEMP(1)
```

```
      FIXS=TEMP(2)
      EN=TEMP(3)
      ES=TEMP(4)
      PN=0.5*T2*EN/SPAN
      PS=0.5*T1*ES/SPAN
      TEMP(1)=TEMP(1)+PN
      TEMP(2)=TEMP(2)+PS
      DO 30 K=5,12
30    TEMP(K)=U(K-4)
      TEMP(13)=PS
      TEMP(14)=PN
      TEMP(15)=FIXS
      TEMP(16)=FIXN
      CALL WRITMS (1,TEMP,17,KEL)
      GO TO 100
C
C
C                        TYPE III ELEMENT
C
35    I=IX(1,KEL)
      J=IX(2,KEL)
      KK=IX(3,KEL)
      L=IX(4,KEL)
      M=IX(5,KEL)
      N=IX(6,KEL)
      MK=ITYP-2
      GO TO (40,45,50), MK
40    I1=2*I-1+NT3
      I2=2*J-1+NT3
      I3=2*KK-1+NT3
      I4=2*L-1+NT3
      I5=2*M-1+NT3
      I6=2*N-1+NT3
      GO TO 55
45    I1=3*I-2
      I2=3*J-2
      I3=2*KK-1+NT3
      I4=3*L-2
      I5=2*M-1+NT3
      I6=2*N-1+NT3
      GO TO 55
50    I1=2*I-1+NT3
      I2=2*J-1+NT3
      I3=3*KK-2
      I4=2*L-1+NT3
      I5=2*M-1+NT3
      I6=2*N-1+NT3
55    CONTINUE
      AA=AREAA(KEL)
      YC=(Y(I)+Y(J)+Y(KK))/3.
      IF (YC.GT.H2) GO TO 100
      CALL READMS (3,DMAT,89,KEL)
      D11=DMAT(1)
      D12=DMAT(2)
      D13=DMAT(3)
      D21=DMAT(4)
      D22=DMAT(5)
      D23=DMAT(6)
      D31=DMAT(7)
      D32=DMAT(8)
```

```
      D33=DMAT(9)
      NUE1=DMAT(10)
      NUE2=DMAT(11)
      X21=X(J)-X(I)
      X13=X(I)-X(KK)
      X32=X(KK)-X(J)
      Y23=Y(J)-Y(KK)
      Y31=Y(KK)-Y(I)
      Y12=Y(I)-Y(J)
      U1=R(I1)
      U2=R(I2)
      U3=R(I3)
      U4=R(I4)
      U5=R(I5)
      U6=R(I6)
      V1=R(I1+1)
      V2=R(I2+1)
      V3=R(I3+1)
      V4=R(I4+1)
      V5=R(I5+1)
      V6=R(I6+1)
C
      KODE=0
      IF (KEL.GT.NT12.AND.KEL.LE.(NT12+NT1/2)) GO TO 60
      GO TO 65
   60 ETATWO=1.-ETA
      ETAFOUR=1.-2.*ETA
      ETA2=ETATWO*ETATWO
      ETA4=ETAFOUR*ETAFOUR
      KODE=1
      GO TO 70
   65 MT=IX(8,KEL)
      ETAFOUR=1.0
      ETATWO=ETAFOUR
      ETA4=1.0
      ETA2=ETA4
C
C        GENERATE AND STORE STRESS AND STRAINS ELEMENT BY ELEMENTC
C
   70 K=KEL
      F1=(4.*ZAI(1)-ETATWO)/ETATWO
      F2=(4.*ZAI(2)-1.)/ETAFOUR
      F3=4.*ZAI(3)-ETAFOUR
C
C                            STRAINS AT CENTROID
C
      STRAINX=Y23*F1*U1+Y31*F2*U2+Y12*F3*U3+(Y23*4.*ZAI(2)+Y31*4.*ZAI
     1(1))*U4/ETA2+(Y31*4.*ZAI(3)+Y12*4.*ZAI(2))*U5/ETAFOUR+(Y23*4.*Z
     2AI(3)+Y12*4.*ZAI(1))*U6/ETAFOUR
      STRAINX=0.5*STRAINX/AA
      STRAINY=X32*F1*V1+X13*F2*V2+X21*F3*V3+4.*(X32*ZAI(2)+X13*ZAI(1)
     1)*V4/ETA2+4.*(X13*ZAI(3)+X21*ZAI(2))*V5/ETAFOUR+4.*(X32*ZAI(3)+
     2X21*ZAI(1))*V6/ETAFOUR
      STRAINY=0.5*STRAINY/AA
C
      A1=X32*F1*U1
      A2=X13*F2*U2
      A3=X21*F3*U3
      A4=4.*(X32*ZAI(2)+X13*ZAI(1))*U4/ETA2
      A5=4.*(X13*ZAI(3)+X21*ZAI(2))*U5/ETAFOUR
```

```
      A6=4.*(X32*ZAI(3)+X21*ZAI(1))*U6/ETAFOUR
      B1=Y23*F1*V1
      B2=Y31*F2*V2
      B3=Y12*F3*V3
      B4=4.*(Y23*ZAI(2)+Y31*ZAI(1))*V4/ETA2
      B5=4.*(Y31*ZAI(3)+Y12*ZAI(2))*V5/ETAFOUR
      B6=4.*(Y23*ZAI(3)+Y12*ZAI(1))*V6/ETAFOUR
      STRANXY=A1+A2+A3+A4+A5+A6+B1+B2+B3+B4+B5+B6
      STRANXY=0.5*STRANXY/AA
C
C
C         ELEMENT STRESSES AT CENTROID IN X Y AND XY DIRECTIONS
C
      STRESSX=D11*STRAINX+D12*STRAINY+D13*STRANXY
      STRESSY=D21*STRAINX+D22*STRAINY+D23*STRANXY
      STRESXY=D31*STRAINX+D32*STRAINY+D33*STRANXY
C
C     DETERMINE PRINCIPLE STRESSES SIGMA1, SIGMA2,SIGMA3 ... ... ...
C
      STRESS1=-(STRESSX+STRESSY)/2.
      STRESS2=SQRT(((STRESSX-STRESSY)/2.)**2+STRESXY**2)
      SIGMA1=STRESS1+STRESS2
      SIGMA3=(STRESS1-STRESS2)
      SIGMA2=(SIGMA1+SIGMA3)*(NUE1+NUE2)/2.
C
C       DETERMINE OCTAHEADRAL STRESSES SGMAOCT, TAUOCT ... ... ...
C
      SGMAOCT=(SIGMA1+SIGMA2+SIGMA3)/3.
      TAUOCT=(SQRT((SIGMA1-SIGMA2)**2+(SIGMA2-SIGMA3)**2+(SIGMA3-SIGM
     1A1)**2))/3.
C
C     DETERMINE PRINCIPLE STRAINS EPLISN1, EPLISN2, EPLISN3 ... ...
C
      E1=(STRAINX+STRAINY)/2.
      E2=SQRT(((STRAINX-STRAINY)/2.)**2+STRANXY**2)
      EPLISN1=E1+E2
      EPLISN3=E1-E2
      EPLISN2=(EPLISN1+EPLISN3)*(NUE1+NUE2)/2.
C
C     DETERMINE OCTAHEADRAL STRAINS - EPSNOCT, GAMAOCT ... ... ...
C
      EPSNOCT=(EPLISN1+EPLISN2+EPLISN3)/3.
      GAMAOCT=2.*(SQRT((EPLISN1-EPLISN2)**2+(EPLISN2-EPLISN3)**2+(EPL
     1ISN3-EPLISN1)**2))/3.
C
C                 STORE STRESSES AND STRAINS ON TAPE 1
C
      CALL READMS (1,TEMP,17,KEL)
      DELSIG1=SIGMA1
      SIG1=TEMP(4)
      STRESSX=STRESSX+TEMP(1)
      STRESSY=STRESSY+TEMP(2)
      STRESXY=STRESXY+TEMP(3)
      SIGMA1=SIGMA1+TEMP(4)
      SIGMA2=SIGMA2+TEMP(5)
      SIGMA3=SIGMA3+TEMP(6)
      SGMAOCT=SGMAOCT+TEMP(7)
      TAUOCT=TAUOCT+TEMP(8)
      STRAINX=STRAINX+TEMP(9)
      STRAINY=STRAINY+TEMP(10)
```

```
      STRANXY=STRANXY+TEMP(11)
      EPLISN1=EPLISN2+TEMP(12)
      EPLISN2=EPLISN2+TEMP(13)
      EPLISN3=EPLISN3+TEMP(14)
      EPSNOCT=EPSNOCT+TEMP(15)
      GAMAOCT=GAMAOCT+TEMP(16)
C
      TEMP(1)=STRESSX
      TEMP(2)=STRESSY
      TEMP(3)=STRESXY
      TEMP(4)=SIGMA1
      TEMP(5)=SIGMA2
      TEMP(6)=SIGMA3
      TEMP(7)=SGMAOCT
      TEMP(8)=TAUOCT
      TEMP(9)=STRAINX
      TEMP(10)=STRAINY
      TEMP(11)=STRANXY
      TEMP(12)=EPLISN1
      TEMP(13)=EPLISN2
      TEMP(14)=EPLISN3
      TEMP(15)=EPSNOCT
      TEMP(16)=GAMAOCT
C
      CALL WRITMS (1,TEMP,17,KEL)
C
      IF (NOTENSN.LE.0) GO TO 100
C
C                           NO-TENSION ANALYSIS
C
      SIGTOTL=SIG1+DELSIG1
      IF (SIGTOTL+GAMA(MT)) 75,100,100
   75 WRITE (6,165) KEL,SIG1,DELSIG1
      CALL ZERO (PP,12)
      CALL NULLMAT (SKJT,12,12)
      RATIO=1.-ABS(SIG1)/ABS(DELSIG1)
      MO=11
      DO 85 IM=1,12
         DO 80 JM=IM,12
            MO=MO+1
   80    SKJT(IM,JM)=SKJT(JM,IM)=DMAT(MO)
   85 CONTINUE
      U(1)=U1
      U(2)=U2
      U(3)=U3
      U(4)=U4
      U(5)=U5
      U(6)=U6
      U(7)=V1
      U(8)=V2
      U(9)=V3
      U(10)=V4
      U(11)=V5
      U(12)=V6
      DO 95 JI=1,12
         SUM=0.0
         DO 90 KI=1,12
   90    SUM=SUM+SKJT(JI,KI)*U(KI)
         PP(JI)=-SUM*RATIO
   95 CONTINUE
```

```
C
         Q(I1)=Q(I1)+PP(1)
         Q(I2)=Q(I2)+PP(2)
         Q(I3)=Q(I3)+PP(3)
         Q(I4)=Q(I4)+PP(4)
         Q(I5)=Q(I5)+PP(5)
         Q(I6)=Q(I6)+PP(6)
         Q(I1+1)=Q(I1+1)+PP(7)
         Q(I2+1)=Q(I2+1)+PP(8)
         Q(I3+1)=Q(I3+1)+PP(9)
         Q(I4+1)=Q(I4+1)+PP(10)
         Q(I5+1)=Q(I5+1)+PP(11)
         Q(I6+1)=Q(I6+1)+PP(12)
         IFLAG=1
C
  100 CONTINUE
      IF (NT3.LE.0) GO TO 160
      IF (INTER.LE.0) GO TO 160
C
C                NO-TENSION ANALYSIS FOR INTERACTION ELEMENT
C
      NSTART=NT1+1
      DO 155 KEL=NSTART,NT12,2
         MT=IX(8,KEL)
         FR(2)=0.0
         FR(1)=FR(2)
         CALL READMS (1,TEMP,17,KEL)
         CALL READMS (1,TEMP,17,KEL+1)
         PNORML=(TEMP(1)+TEMP(1))*0.5
         PSHEAR=(TEMP(2)+TEMP(2))*0.5
         IF (ABS(PNORML).LT.GAMA(MT)) GO TO 155
         IF (PNORML+GAMA(MT)) 110,105,105
  105    IF (NCYCLE.EQ.1) GO TO 155
         RATIO=ABS(PSHEAR/PNORML)
         IF (RATIO.LE.SLIP) GO TO 155
         R1=ABS(TEMP(15)/TEMP(16))
         QQ=TEMP(13)/TEMP(14)
         FIXN=TEMP(16)
         PN=TEMP(14)
         DR=SLIP-R1
         T3=QQ-R1-DR
         DSR=DR*FIXN/T3
         T4=(PN-DSR)/PN
         FR(1)=T4+FR(1)
         FR(1)=-FR(1)
         WRITE (6,170) KEL,RATIO,SLIP,R1,QQ,DR,DSR,FIXN,PN,FR(1)
         R1=ABS(TEMP(15)/TEMP(16))
         QQ=TEMP(13)/TEMP(14)
         FIXN=TEMP(16)
         PN=TEMP(14)
         DR=SLIP-R1
         T3=QQ-R1-DR
         DSR=DR*FIXN/T3
         T4=(PN-DSR)/PN
         FR(2)=T4+FR(2)
         FR(2)=-FR(2)
         KEL1=KEL+1
         WRITE (6,170) KEL1,RATIO,SLIP,R1,QQ,DR,DSR,FIXN,PN,FR(2)
         GO TO 115
C
```

```
  110     T1=(TEMP(16)+TEMP(16))/2.
          T2=(TEMP(14)+TEMP(14))/2.
C
C                       FR(1)=FR(2)=1.-ABS(T1)/ABS(T2)
C
          T3=1.-T1/T2
          FR(2)=T3/2.
          FR(1)=FR(2)
          KEL1=KEL+1
          WRITE (6,175) KEL,KEL1,FR(1),FR(2)
  115     NUM=KEL
          NO=1
  120     CALL READMS (7,DMAT,64,NUM)
          MO=0
          DO 125 IM=1,8
          DO 125 JM=1,8
             MO=MO+1
             SKJT(IM,JM)=DMAT(MO)
  125     CONTINUE
          DO 140 IM=1,8
             GO TO (130,135), NO
  130        U(IM)=TEMP(IM+4)
             GO TO 140
  135        U(IM)=TEMP(IM+4)
  140     CONTINUE
          DO 150 JI=1,8
             SUM=0.0
             DO 145 KI=1,8
  145        SUM=SUM+SKJT(JI,KI)*U(KI)
C
C                             PP(JI)=-SUM * FR(NO)
C
             PP(JI)=SUM*FR(NO)
  150     CONTINUE
          WRITE (6,180) (PP(JI),JI=1,8)
          I1=IA(NUM,1)
          I2=IA(NUM,2)
          I3=IA(NUM,3)
          I4=IA(NUM,4)
C
          Q(I1)=Q(I1)+PP(1)
          Q(I1+1)=Q(I1+1)+PP(2)
          Q(I2)=Q(I2)+PP(3)
          Q(I2+1)=Q(I2+1)+PP(4)
          Q(I3)=Q(I3)+PP(5)
          Q(I3+1)=Q(I3+1)+PP(6)
          Q(I4)=Q(I4)+PP(7)
          Q(I4+1)=Q(I4+1)+PP(8)
          IFLAG=1
          NO=NO+1
          IF (NO.GT.2) GO TO 155
          NUM=KEL+1
          GO TO 120
  155 CONTINUE
  160 WRITE (6,185)
      RETURN
C
  165 FORMAT (10X, 24HTENSION IN ELEMENT NO = ,I5,  8H SIGMA1=,E12.3, 13
     1H DELSIGMA1 = ,E12.3)
  170 FORMAT (1X,  8HINT SLIP,I5,9E12.3)
```

```
  175 FORMAT (1X, 20HINTERACTION  TENSION,2I5,2E12.3)
  180 FORMAT (1X,  9HINT.FORCE,8E12.3)
  185 FORMAT (1X, 23HOVERLAY (4,0) COMPLETED)
C
      END
      OVERLAY(ROY,5,0)
C                                OVERLAY(ROY,5,0)
      PROGRAM RESULTS
      COMMON NNODES,NELEMNT,NDOF,MBAND,ND,NT3,ISTOP,NCYCLE,LAYERS,ISTEP,
     1NSTEP,NT12,ETA,NT1,NT2,NUTENSN,IFLAG,NSIZE,NCODE(550),X(550),Y(550
     2),JNDX(51),ANLSIS,IX(8,250),AREAA(250),INDX(250),INDEX(250),GAMA(2
     35),ZAI(3)
      COMMON /1/ E,NUE,RADIUS,XCEN,YCEN,EI,KN,KS,H1,H2,INTER
      COMMON /3/ R
      DIMENSION TEMP(17), R(1100), RR(1100)
      DIMENSION E(10), NUE(10)
      DIMENSION TMP(36), U(6), SRG(6,6), PXY(30), QXY(30), MOM(30), P(30
     1), Q(30), PP(6)
      DIMENSION SG(30), S1(90)
      REAL NUE
      REAL KN,KS
      REAL MOM
      INTEGER U1,U2,U3,U4,U5,U6,U7,U8,U9
C
C     THIS ROUTINE RECORDS RESULTS FROM DIFFERENT ROUTINES AND TAPES AND
C       STORES, ADDS, FOR SUCCESSIVE TERMS IF INCREMENTAL ANALYSIS AND
C                                FINALLY OUTPUTTED
C
C     WRITTING NODAL FORCES, DISPLACEMENTS, ELEMENT STRESSES, STRAINS ..
C
      REWIND 2
      NDRING=3*NT3
      DO 5 I=1,NDRING
    5 S1(I)=R(I)
      READ (2) (RR(I),I=1,NDOF)
      DO 10 I=1,NDOF
         R(I)=R(I)+RR(I)
   10 CONTINUE
      REWIND 2
      WRITE (2) (R(I),I=1,NDOF)
C
      WRITE (6,90) ISTEP,NCYCLE
      IF (NT3.LE.0) GO TO 20
      WRITE (6,95)
      DO 15 I=1,NT3
         K1=3*I-2
         WRITE (6,100) I,R(K1),R(K1+1),R(K1+2)
   15 CONTINUE
   20 IF (NCYCLE.NE.NSTEP) GO TO 35
      IF (IFLAG.GT.0) GO TO 35
      WRITE (6,105)
      NT3P1=NT3+1
      DO 25 I=NT3P1,NNODES,3
         K1=2*I-1+NT3
         K2=K1+1
         K3=2*(I+1)-1+NT3
         K4=K3+1
         K5=2*(I+2)-1+NT3
         K6=K5+1
         I2=I+2
```

```
      WRITE (6,110) I,R(K1),R(K2),I1,R(K3),R(K4),I2,R(K5),R(K6)
   25 CONTINUE
C
C                        PRINT ELEMENT STRESSES
C
      WRITE (6,115)
      NT12P1=NT12+1
      DO 30 I=NT12P1,NELEMNT
         CALL READMS (1,TEMP,17,I)
         WRITE (6,120) (I,(TEMP(J),J=1,8))
   30 CONTINUE
C
C        CALCULATION OF NORMAL, SHEAR FORCE AND MOMENTS FOR PIPE NODES
C
   35 IF (NT3.LE.0) GO TO 85
      WRITE (6,125)
      NEL=NT3-1
      DO 55 I=1,NEL
         IF (IX(7,I).NE.1) GO TO 55
         I1=IX(1,I)
         I2=IX(2,I)
         U1=3*(I1-1)+1
         U2=U1+1
         U3=U1+2
         U4=3*(I2-1)+1
         U5=U4+1
         U6=U4+2
         U(1)=S1(U1)
         U(2)=S1(U2)
         U(3)=S1(U3)
         U(4)=S1(U4)
         U(5)=S1(U5)
         U(6)=S1(U6)
         CALL READMS (7,TMP,36,I)
C
         MO=0
         CALL READMS (1,TEMP,17,I)
         DO 40 K=1,6
         DO 40 L=1,6
            MO=MO+1
            SRG(K,L)=TMP(MO)
   40    CONTINUE
         DO 50 K=1,6
            SUM=0.0
            DO 45 L=1,6
   45       SUM=SUM+SRG(K,L)*U(L)
            PP(K)=SUM
   50    CONTINUE
         PXY(I1)=PP(4)+TEMP(3)
         QXY(I1)=PP(5)+TEMP(4)
         MOM(I1)=PP(6)+TEMP(5)
         PXY(I2)=PP(1)+TEMP(6)
         QXY(I2)=PP(2)+TEMP(7)
         MOM(I2)=PP(3)+TEMP(8)
         TEMP(3)=PXY(I1)
         TEMP(4)=QXY(I1)
         TEMP(5)=MOM(I1)
         TEMP(6)=PXY(I2)
         TEMP(7)=QXY(I2)
         TEMP(8)=MOM(I2)
```

```
         CALL WRITMS (1,TEMP,17,I)
   55 CONTINUE
      DO 70 I=1,NT3
         IF (I.EQ.NT3) GO TO 60
         GO TO 65
   60    PXY(I)=-PXY(I)
         QXY(I)=-QXY(I)
         MOM(I)=-MOM(I)
   65    YC=Y(I)-YCEN
         XC=X(I)-XCEN
         IF (XC.EQ.0.0) XC=0.00001
         TN=YC/XC
         THETA=ATAN(TN)
         SN=SIN(THETA)
         CS=COS(THETA)
         P(I)=-PXY(I)*SN+QXY(I)*CS
         Q(I)=-PXY(I)*CS+QXY(I)*SN
         WRITE (6,130) I,P(I),Q(I),MOM(I)
   70 CONTINUE
      WRITE (6,140)
      NSTART=NT1+1
      DO 75 I=NSTART,NT12,2
         CALL READMS (1,TEMP,17,I)
         T1=TEMP(1)
         T2=TEMP(2)
         CALL READMS (1,TEMP,17,I+1)
         TEMP(1)=0.5*(T1+TEMP(1))
         TEMP(2)=0.5*(T2+TEMP(2))
         RATIO=ABS(TEMP(2)/TEMP(1))
         IP1=I+1
         WRITE (6,135) I,IP1,TEMP(1),TEMP(2),RATIO
   75 CONTINUE
C
C                              PRINT ELEMENT STRAINS
C
      WRITE (6,145)
      DO 80 I=NT12P1,NELEMNT
         CALL READMS (1,TEMP,17,I)
         WRITE (6,120) (I,(TEMP(J),J=9,16))
   80 CONTINUE
   85 WRITE (6,150)
      RETURN
C
   90 FORMAT (1H1,10X, 14HRESULTS AFTER ,I5, 17H   LAYERS OF FILL,5X, 15
     1HINCREMENT NO = ,I5//)
   95 FORMAT (10X, 28H3 - D.O.F. NODAL DEFLECTIONS,//10X,  8HNODE NO.,5X
     1, 14HX - DEFLECTION,5X, 14HY - DEFLECTION,5X,  8HROTATION,/)
  100 FORMAT (7X,I5,7X,3(E14.4,5X))
  105 FORMAT (//10X, 21HNODE POINT DEFLECTION,//3(3X,  8HNODE NO.,3X, 12
     1HX-DEFLECTION,3X, 12HY-DEFLECTION)//)
  110 FORMAT (3(5X,I4,5X,E12.4,3X,E12.4))
  115 FORMAT (1H1/10X, 16HELEMENT STRESSES,//5X, 11HELEMENT NO.,3X,  7HS
     1TRESSX,7X,  7HSTRESSY,7X,  7HSTRESXY,7X,  6HSIGMA1,8X,  6HSIGMA2,8
     2X,  6HSIGMA3,7X,  8HSIGMAOCT,6X,  7HTAUOCTA,/)
  120 FORMAT (5X,I5,2X,8E14.2)
  125 FORMAT (//10X, 47HNORMAL, AND SHEAR FORCES, MOMENTS AT PIPE NODES,
     1/10X,  8HNODE NO.,10X, 12HNORMAL FORCE,5X, 11HSHEAR FORCE,5X, 12HN
     2ODAL MOMENT,/)
  130 FORMAT (13X,I3,12X,E12.2,2(4X,E12.2))
  135 FORMAT (3X,I2,  3H + ,I2,12X,E10.2,12X,E10.2,12X,F10.4)
```

```
  140 FORMAT (///10X, 52HNORMAL AND SHEAR STRESS ON SOIL IN INTERACTION
     1LAYER,//5X,  7HELE.NO.,10X, 12HNORMAL STRS.,10X, 12HSHEAR STRESS,1
     20X, 12H RATIO T/N   ,/)
  145 FORMAT (1H1/10X, 15HELEMENT STRAINS,//5X, 11HELEMENT NO.,3X,  7HST
     1RAINX,7X,  7HSTRAINY,7X,  7HSTRANXY,7X,  7HEPSILN1,7X,  7HEPSILN2,
     27X,  7HEPSILN3,7X,  7HEPSNOCT,7X,  7HGAMAOCT,/)
  150 FORMAT (1X, 23HOVERLAY (5,0) COMPLETED)
C
      END
      OVERLAY(ROY,6,0)
C                                     OVERLAY(ROY,6,0)
      PROGRAM RESOLV
      COMMON NNODES,NELEMNT,NDOF,MBAND,ND,NT3,ISTOP,NCYCLE,LAYERS,ISTEP,
     1NSTEP,NT12,ETA,NT1,NT2,NOTENSN,IFLAG,NSIZE,NCODE(550),X(550),Y(550
     2),JNDX(51),ANLSIS,IX(8,250),AREAA(250),INDX(250),INDEX(250),GAMA(2
     35),ZAI(3)
      COMMON /3/ R
      COMMON /5/ Q(1100),LIST(1101)
      DIMENSION A(103,206), B(206), ARRAY(103)
      DIMENSION R(1100)
      REWIND 9
C
C     SOLUTION OF STIFFNESS EQUATIONS FOR GIVEN BOUNDARY CONDITIONS AND
C
      NUMBLK=1
      NL=1
      NM=ND
      KSHIFT=0
      ND2=2*ND
      NEND=3*NT3
      NDINC=1
      IF (NT3.LE.0) NDINC=0
      IF (NT3.LE.0) ND=ND-1
      CALL NULLMAT (A,ND,ND2)
      CALL ZERO (B,ND2)
      NX=NL-1
      CALL ZERO (ARRAY,NSIZE)
      DO 10 N=1,NSIZE
         CALL READMS (10,ARRAY,NSIZE,NX+N)
         DO 5 M=1,NSIZE
    5    A(N,M)=ARRAY(M)
   10 CONTINUE
      GO TO 30
   15 NX=NL-1
      CALL ZERO (ARRAY,NSIZE)
      DO 25 N=1,NSIZE
         CALL READMS (10,ARRAY,NSIZE,NX+N)
         DO 20 M=1,NSIZE
   20    A(N,M)=A(N,M)+ARRAY(M)
   25 CONTINUE
C
C                  MODIFICATION FOR LOAD AND BOUNDARY CONDITIONS
C
   30 I=NL
   35 IF (I.GT.NEND) GO TO 70
      FIND=FLOAT(I+2)/3.
      II=INT(FIND)
      PX=Q(I)
      PY=Q(I+1)
      ICODE=NCODE(II)-3
```

```
      GO TO (40,45,50,55,60,65), ICODE
   40 J=I-KSHIFT
      B(J)=B(J)+PX
      B(J+1)=B(J+1)+PY
      I=I+3
      GO TO 35
   45 J=I-KSHIFT
      P=0.0
      CALL MODIFY (J,ND,P,A,B)
      B(J+1)=B(J+1)+PY
      I=I+3
      GO TO 35
   50 J=I-KSHIFT
      B(J)=B(J)+PX
      P=0.0
      CALL MODIFY (J+1,ND,P,A,B)
      I=I+3
      GO TO 35
   55 J=I-KSHIFT
      B(J)=B(J)+PX
      B(J+1)=B(J+1)+PY
      P=0.0
      CALL MODIFY (J+2,ND,P,A,B)
      I=I+3
      GO TO 35
   60 J=I-KSHIFT
      P=0.0
      CALL MODIFY (J,ND,P,A,B)
      B(J+1)=B(J+1)+PY
      CALL MODIFY (J+2,ND,P,A,B)
      I=I+3
      GO TO 35
   65 J=I-KSHIFT
      B(J)=B(J)+PX
      P=0.0
      CALL MODIFY (J+1,ND,P,A,B)
      CALL MODIFY (J+2,ND,P,A,B)
      I=I+3
      GO TO 35
   70 IF (I.GT.NM.OR.I.GT.NDOF) GO TO 95
      FIND=FLOAT(I-NT3+1)/2.
      II=INT(FIND)
      PX=Q(I)
      PY=Q(I+1)
      ICODE=NCODE(II)+1
      GO TO (75,80,85,90), ICODE
   75 J=I-KSHIFT
      B(J)=B(J)+PX
      B(J+1)=B(J+1)+PY
      I=I+2
      GO TO 70
   80 J=I-KSHIFT
      P=0.0
      CALL MODIFY (J,ND,P,A,B)
      B(J+1)=B(J+1)+PY
      I=I+2
      GO TO 70
   85 J=I-KSHIFT
      B(J)=B(J)+PX
      P=0.0
```

```
      CALL MODIFY (J+1,ND,P,A,B)
      I=I+2
      GO TO 70
   90 J=I-KSHIFT
      P=0.0
      CALL MODIFY (J,ND,P,A,B)
      CALL MODIFY (J+1,ND,P,A,B)
      I=I+2
      GO TO 70
   95 CONTINUE
C
C                          REDUCE BLOCK OF EQUATIONS
C
      DO 110 N=1,ND
         IF (A(1,N).EQ.0.0) GO TO 110
         B(N)=B(N)/A(1,N)
         DO 105 L=2,MBAND
            IF (A(L,N).EQ.0.0) GO TO 105
            CX=A(L,N)/A(1,N)
            I=N+L-1
            J=0
            DO 100 K=L,MBAND
               J=J+1
  100       A(J,I)=A(J,I)-CX*A(K,N)
            B(I)=B(I)-A(L,N)*B(N)
            A(L,N)=CX
  105    CONTINUE
  110 CONTINUE
      IF (NM.GE.NDOF) GO TO 120
C
C                WRITE BLOCK OF EQUATIONS (REDUCED) ON TAPE 9
C
      WRITE (9) (B(N),(A(M,N),M=1,MBAND),N=1,ND)
C
C                     SHIFT BLOCK OF EQN. UP FOR NEXT BLOCK
C
      DO 115 N=1,ND
         MM=ND+N
         B(N)=B(MM)
         B(MM)=0.0
      DO 115 M=1,MBAND
         A(M,N)=A(M,MM)
         A(M,MM)=0.0
  115 CONTINUE
      KSHIFT=KSHIFT+ND
      IF (NUMBLK.EQ.1) ND=ND-NDINC
      NUMBLK=NUMBLK+1
      NM=NM+ND
      NL=NM-ND+1
      GO TO 15
C
C               BACK SUBSTITUTION IN GAUSS ELIMINATION PROCESS
C
  120 CALL ZERO (R,NDOF)
      IF (NUMBLK.EQ.1) NDINC=0
C
      NU=ND*NUMBLK+1+NDINC
      NB=NUMBLK
  125 DO 135 M=1,ND
         N=ND+1-M
```

```
         DO 130 K=2,MBAND
            L=N+K-1
  130    B(N)=B(N)-A(K,N)*B(L)
         NM=N+ND
         IF (NB.EQ.2) NM=NM+NDINC
         B(NM)=B(N)
         NU=NU-1
  135 R(NU)=B(N)
      NB=NB-1
      IF (NB.EQ.1) ND=ND+NDINC
      IF (NB.LE.0) GO TO 140
      BACKSPACE 9
      READ (9) (B(N),(A(M,N),M=1,MBAND),N=1,ND)
      BACKSPACE 9
      GO TO 125
  140 CONTINUE
      RETURN
C
C
      END
      SUBROUTINE MODIFY (N,ND,P,A,B)
      DIMENSION A(103,206), B(206)
C
C                    MODIFICATION FOR BOUNDARY CONDITIONS
C
      ND2=2*ND
      DO 10 M=2,ND
         K=N-M+1
         IF (K.LE.0) GO TO 5
         B(K)=B(K)-A(M,K)*P
         A(M,K)=0.0
    5    K=N+M-1
         IF (K.GT.ND2) GO TO 10
         B(K)=B(K)-A(M,N)*P
         A(M,N)=0.0
   10 CONTINUE
      A(1,N)=0.0
      B(N)=0.0
      RETURN
C
      END
```

APPENDIX - II

PROPRTY
PROGRAM LISTING

```
      PROGRAM PROPRTY (INPUT,OUTPUT)
      DIMENSION XP(20), YP(20), VP(20), YDP1(20), YDPV(20), TITL(10)
      REAL NUE,NUEIN
      DATA PLSTRS,PLSTRN/6HPLSTRS,6HPLSTRN/
C
C       THIS PROGRAM GENERATES DATA REQUIRED FOR EVALUATION OF TANGENT
C      MODULUS AND POISSONS RATIO VS. SIGMA(OCT) FOR VARIOUS STRESS RATI
C
      READ 70, (TITL(I),I=1,10)
      READ 75, TEST,NOCURVS,RF
      IF (TEST.EQ.PLSTRS) NGO=1
      IF (TEST.EQ.PLSTRN) NGO=2
      DO 65 IX=1,NOCURVS
         READ 80, SIGMA3,NP
         READ 85, (XP(I),YP(I),VP(I),I=1,NP)
         PRINT 70, (TITL(I),I=1,10)
         PRINT 90, IX,SIGMA3,NP,RF
         PRINT 95, (I,XP(I),YP(I),VP(I),I=1,NP)
         DO 5 I=1,NP
    5    VP(I)=VP(I)-XP(I)
         S1IN=SIGMA3
         XIN=0.0
         E3IN=0.0
         CALL SPLINE (NP,XP,YP,YDP1)
         CALL SPLINE (NP,XP,VP,YDPV)
         PRINT 100
         X=0.00001
         DX=0.000001
         NUE=0.5
   10    DO 15 L=2,NP
            IF (X.LT.XP(L).AND.X.GE.XP(L-1)) GO TO 20
   15    CONTINUE
         GO TO 65
   20    A=YDP1(L-1)
         B=YDP1(L)
         C=XP(L-1)
         D=XP(L)
         E=YP(L-1)
         F=YP(L)
         PP=X
         S1=ORDINET(A,B,C,D,E,F,PP)
C
         DO 25 L=2,NP
            IF (X.LT.XP(L).AND.X.GE.XP(L-1)) GO TO 30
   25    CONTINUE
         GO TO 65
   30    A=YDPV(L-1)
         B=YDPV(L)
         C=XP(L-1)
         D=XP(L)
         E=VP(L-1)
         F=VP(L)
         PP=X
         E3=ORDINET(A,B,C,D,E,F,PP)
         B1=S1+SIGMA3
         B3=SIGMA3
         GO TO (35,40), NGO
   35    B2=B3
         GO TO 45
```

```
   40    B2=NUE*(B1+B3)
   45    SGMAOCT=(B1+B2+B3)/3.
         A1=B1-B2
         A2=B2-B3
         A3=B3-B1
         TAUOCT=0.333*SQRT(A1**2+A2**2+A3**2)
         T1=X
         T3=E3
         GO TO (50,55), NGO
   50    T2=T3
         GO TO 60
   55    T2=0.0
   60    A1=T1-T2
         A2=T2-T3
         A3=T3-T1
         GAMAOCT=2.*(SQRT(A1**2+A2**2+A3**2))/3.
         TAUF=RF*SGMAOCT
         RATIO=TAUOCT/TAUF
         EP3=E3
         T1=SIGMA3*X-B1*EP3
         T2=(SIGMA3+B1)*(X-EP3)
         NUE=T1/T2
         EM1=B1*(1.-NUE**2)/X
         EM2=NUE*(1.+NUE)*SIGMA3/X
         EM=EM1-EM2
         EM=100.*EM
         DS1=S1-S1IN
         DEP1=X-XIN
         DEP3=EP3-E3IN
         T1=DS1/(DEP1-DEP3)
         T2=DEP3/(DEP1-DEP3)
         TMOD=T1*(1.-T2)
         TMOD=100.*TMOD
         TANNUE=-DEP3/(DEP1-DEP3)
         PRINT 105, X,E3,S1,TAUOCT,GAMAOCT,NUE,EM,TANNUE,SGMAOCT,TMOD,RA
     1   TIO
         XIN=X
         S1IN=S1
         E3IN=EP3
         DX=1.1*DX
         X=X+DX
         IF (X.GT.XP(NP)) GO TO 65
         GO TO 10
   65 CONTINUE
      STOP
C
   70 FORMAT (10A8)
   75 FORMAT (A6,I3,F10.0)
   80 FORMAT (5X,F10.0,I5)
   85 FORMAT (3F5.0)
   90 FORMAT (10X, 12HCURVE NO. = ,I5, 11H  SIGMA3 = ,F8.2, 16HNO. OF PO
     1INTS = ,I5, 18H  FAILURE RATIO = ,F8.2/10X, 10HINPUT DATA,/2X,  6H
     2SL.NO.,4X,  7HSTRAIN1,8X,  5HS1-S3,6X,  7HSTRAIN3,3X, 11H( PERCENT
     3 ))
   95 FORMAT (4X,I2,4X,3E12.3)
  100 FORMAT (3X,  7HSTRAIN1,5X,  7HSTRAIN3,5X,  6HSIGMA1,7X,  6HTAUOCT,
     16X,  7HGAMAOCT,6X,  3HNUE,7X,  7HMODULUS,4X,  6HTANNUE,4X,  8HSIGM
     2AOCT,6X,  6HTANMOD,6X,  5HRATIO,/)
  105 FORMAT (11E12.4)
C
```

```
      END
      SUBROUTINE SPLINE (NPN,XP,YP,YDP)
      DIMENSION XP(20), YP(20), YDP(20), H(20), AI(20), BI(20), CI(20),
     1DI(20)
      NP1=NPN-1
      DO 5 M=1,NP1
    5 H(M)=XP(M+1)-XP(M)
      SLOP1=FD(H(1),H(2),YP(1),YP(2),YP(3))
      SLOPN=BD(H(NPN-2),H(NPN-1),YP(NPN-2),YP(NPN-1),YP(NPN))
      CALL COFRIT (NPN,XP,YP,SLOP1,SLOPN,AI,BI,CI,DI)
      CALL TRIDGNL (NPN,AI,BI,CI,DI,YDP)
      RETURN
C
      END
      SUBROUTINE COFRIT (NPN,XP,YP,SLOP1,SLOPN,AI,BI,CI,DI)
      DIMENSION XP(20), YP(20), AI(20), BI(20), CI(20), DI(20)
      AI(1)=0.0
      BI(1)=(XP(2)-XP(1))/3.
      CI(1)=BI(1)/2.
      DI(1)=(YP(2)-YP(1))/(XP(2)-XP(1))-SLOP1
      AI(NPN)=(XP(NPN)-XP(NPN-1))/6.
      BI(NPN)=AI(NPN)*2.
      CI(NPN)=0.0
      DI(NPN)=-(YP(NPN)-YP(NPN-1))/(XP(NPN)-XP(NPN-1))+SLOPN
      N1=NPN-1
      DO 5 I=2,N1
         AI(I)=(XP(I)-XP(I-1))/6.
         BI(I)=(XP(I+1)-XP(I-1))/3.
         CI(I)=(XP(I+1)-XP(I))/6.
         DI(I)=(YP(I+1)-YP(I))/(XP(I+1)-XP(I))-(YP(I)-YP(I-1))/(XP(I)-XP
     1   (I-1))
    5 CONTINUE
      RETURN
C
      END
      SUBROUTINE TRIDGNL (NPN,AI,BI,CI,DI,YDP)
      DIMENSION AI(20), BI(20), CI(20), DI(20), YDP(20), Q(30), U(30)
      P=BI(1)
      Q(1)=-CI(1)/P
      U(1)=DI(1)/P
      DO 5 K=2,NPN
         P=AI(K)*Q(K-1)+BI(K)
         Q(K)=-CI(K)/P
         U(K)=(DI(K)-AI(K)*U(K-1))/P
    5 CONTINUE
      YDP(NPN)=U(NPN)
      N1=NPN-1
      DO 10 L=1,N1
         K=N1+1-L
         YDP(K)=Q(K)*YDP(K+1)+U(K)
   10 CONTINUE
      RETURN
C
      END
      FUNCTION ET(R1,R2,Z1,Z2,Z3,S1,S2)
      HJ=Z2-Z1
      T1=-0.5*(S1*(Z2-Z3)**2)/HJ
      T2=0.5*(S2*(Z3-Z1)**2)/HJ
      T3=(R2-R1)/HJ
      T4=-(S2-S1)*HJ/6.
```

```
      ET=T1+T2+T3+T4
      RETURN
C
      END
      FUNCTION FD(S1,S2,R1,R2,R3)
      IF (S1-S2) 5,10,5
    5 FD=(R2-R1)/S1
      RETURN
   10 FD=(-3.*R1+4.*R2-R3)/(2.*S1)
      RETURN
C
      END
      FUNCTION BD(S1,S2,R1,R2,R3)
      IF (S1-S2) 5,10,5
    5 BD=(R3-R2)/S2
      RETURN
   10 BD=(3.*R3-4.*R2+R1)/(2.*S1)
      RETURN
C
      END
      FUNCTION ORDINET(A,B,C,D,E,F,P)
      HJ=D-C
      B1=D-P
      B2=P-C
      A1=B1**3
      A2=B2**3
      T1=A1*A/(6.*HJ)
      T2=A2*B/(6.*HJ)
      T3=(E-A*HJ**2/6.)*(D-P)/HJ
      T4=(F-B*HJ**2/6.)*(P-C)/HJ
      ORDINET=T1+T2+T3+T4
      RETURN
C
      END
```

COVER DESIGN BY ALDO GIORGINI